A COMPLETE INTERACTIVE GRAPHICAL COMPUTER-AIDED INSTRUCTION SYSTEM

Steven Selby Abrams

United States Naval Postgraduate School

THESIS

A COMPLETE INTERACTIVE GRAPHICAL COMPUTER-AIDED INSTRUCTION SYSTEM

by

Steven Selby Abrams

June 1971

A Complete Interactive Graphical Computer-Aided

Instruction System

by

Steven Selby Abrams
Lieutenant Commander, United States Navy
B.S., University of South Carolina, 1963

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
June 1971

ABSTRACT

The use of interactive graphics in computer-aided instruction systems is discussed with emphasis placed on two requirements of such a system. The first is the need to provide the teacher with a useful tool with which to design and modify teaching sessions tailored to the individual needs and capabilities of the students. The second is the requirement to provide for sufficiently flexible interaction with the student during the actual teaching session. These concepts are implemented in a system which maximizes the use of interactive graphics. Facilities available in the system present material to the student in a dynamic, graphical fashion which leads to better understanding and a more enjoyable learning experience. The implemented system, while not yet complete, indicates that a complete working system of this type is feasible and would prove itself to be a valuable educational tool.

TABLE OF CONTENTS

I. INTRODUCTION

Computer-aided instruction (CAI) is defined here to mean the use of computers to aid teachers and students in the educational process, utilizing such functions as presenting material and problems, guiding a student's thinking by asking and answering questions, evaluating his performance, managing a student's path through a course by selecting the material to be presented or by assigning tasks to be completed away from the computer, or any combination of these functions.

Various computer-aided instruction systems have been in limited use for about a decade. To date, there are no generalizable results from controlled experiments which indicate that CAI is more effective than other teaching methods, although it is generally agreed by those in the field that CAI can potentially achieve educational objectives more efficiently and more effectively than other teaching methods. However, two factors make the further development and use of CAI highly desirable if not inevitable. They are the tremendously rapid growth in the volume of human knowledge and the enormous increase in the number of students to be taught.

There are two general approaches to computer-aided instruction. The first is the machine-directed approach in which the various alternatives and paths through an instructional area are programmed into the machine. In this approach there must be a predetermined model of the needs of the student and existing methods by which the computer can evaluate the student's current state of knowledge. In addition, the machine must maintain a record of the student's

past performance and have at its disposal techniques for prescribing a future path for each student which would optimize his learning process.

The second general approach is the student-directed approach in which the sequence of materials presented is altered only at the request of the student. Included in this approach are such things as information retrieval and library functions, learning by discovery and experimentation, simulation and gaming, data reduction, and conversational computing. Some work has been done in the information retrieval and conversational computing aspects of this area of computer-aided instruction, but in general much less has been done and much less is known about this type of approach. Learning by discovery and experimentation has been almost untouched by current research.

This paper deals primarily with the machine-directed CAI approach. A complete interactive graphical computer-aided instruction system is discussed which would provide the teacher with a powerful and flexible tool with which to prepare individualized teaching sessions for each student and receive feedback from the computer describing the students' progress. Each student would be presented material tailored to his needs and achievement level. His progress would be monitored continuously by the machine and a record of his current status in specific instructional areas maintained. New instructional material, review material, drill and practice sessions, tutoring, or other techniques would be utilized depending upon the student's understanding or lack of understanding of specific concepts.

The implementation of one example of a complete interactive graphical computer-aided instruction system is presented. This example is one which uses an Adage AGT-10 graphics terminal interfaced with an XDS 9300 computer to teach basic mathematics on the fourth grade level. Although actual implementation of the system is not complete, sufficient progress has been made to indicate that the system has the potential of becoming a valuable educational tool.

II. BACKGROUND

A. THE NEED FOR COMPUTER-AIDED INSTRUCTION

Three basic educational factors accentuate the need for computer-aided instruction. The first is the trend toward individualized instruction. It has long been recognized by those in the field of education that the fewer the number of students for which a teacher is responsible, the more time and individual attention each student will receive and therefore theoretically the more each student will gain from his educational experience. The logical extention of this philosophy is to advocate individual tutors for each student. However, with the tremendous yearly increase in the number of students coupled with the overall recent trend toward fewer qualified teachers, the prospect of individualized instruction on a person-to-person basis is remote. Conventional mass media such as books, films, television, and even programmed instruction are inherently incapable of individualization. Books, films, and television programs would have to be rewritten to fit each student, and conventional programmed instruction systems lack the flexibility and decision-making capability required of a really effective individualized system. However, CAI seems to have the potential to meet these requirements.

The second factor is the growth in information to be acquired. It has been estimated that human knowledge once doubled only about every 2,000 years but now doubles in less than 10 years. Some significant changes in educational techniques are required in order to dissiminate such vast amounts of information. The growth in

information includes, however, advances in the field of education. Some of this increased knowledge is in areas such as tutoring techniques which will better enable us to implement effective computer-aided instruction systems.

The shortage of qualified teachers is the third factor. Although in recent years, primarily because of short-range economic conditions, the country has apparently had an excess of teachers, it is felt that this condition is only temporary and that once the required expansion in the field of education is continued after the current brief delay, the need for qualified teachers will again be critical. Robert W. Sarnoff noted that in a few years every third college graduate will have to become a teacher if the present pupil-teacher ratio is maintained.

B. THE POTENTIAL OF CAI

Stolurow sees in CAI the capability of (1) individualizing instruction, (2) doing research on teaching under controlled conditions with the ability to collect detailed records of student performance, and (3) developing ways of assisting authors in the development of instructional materials. Applications, aside from instruction, include the development of teaching models, curriculum planning, man-machine relations, and evaluation of student performance. R. W. Gerald lists these benefits of computer-aided instruction: (1) better and faster learning since the student can time his learning at his convenience, go at his own pace, and catch up on missed work if necessary; (2) better teaching at many levels

and in many areas, (3) personalized tutoring; (4) automatic measuring of progress; and (5) the opportunity to work with vastly richer materials and more sophisticated problems. For the teacher, the system (1) takes away a great deal of drudgery and repetition, (2) allows him to be updated effectively, (3) encourages frequent changes in the actual material used, and (4) makes more time available for teacher-student contact.

In general, it is felt that computer-aided instruction is, and will increasingly be, a valuable asset to the field of education, not as a replacement for the human teacher, but rather as a tool which will enable the teacher to accomplish his task more effectively and efficiently. From the standpoint of the student, CAI promises a welcomed expansion of horizons in terms of additional subject matter available and greater diversity in the presentation of the subject matter.

C. CURRENT STATE OF THE ART

Computer-aided instruction is an indirect decendent of the old programmed instruction (PI) concept in which the student works at his own pace on a program with a book, machine, or some other device. The program is linear in nature in that usually there is a predetermined sequence of frames which are presented for all students regardless of the various students' individual differences. Even when branching is allowed, the decision to branch to another set of frames or to continue on the current path is based on insufficient data, thus severely limiting the value of the branching

as an attempt toward individualization. The main disadvantages of PI are that the instructional sequence is not truly individualized since the material presented is not dependent upon the responses and current techniques of frame writing lead to redundancy, making exact knowledge of results by the student seldom necessary.

The true individualization of instruction and the diversity of presentation techniques represent the real departure of CAI from PI. All material presented can be strictly tailored to each student's overall abilities, current achievement level and personal learning characteristics and preferences. In addition, enough diversity can be included in a CAI teaching session to prevent boredum and to obtain and maintain the students' attention and interest.

Currently there are many effective computer-aided instruction systems in use or in the development stage which meet some of the specified educational requirements. One such system is IBM's Coursewriter. It is an interpreter language consisting of about twelve executable instructions and ten manipulative commands. With it an author at an IBM 1050 terminal can enter and edit text material and branching logic onto disk storage. The edit commands include insert, delete, and type and reference text by line number. In operation with a student at a 1050 terminal the stored instructions are interpreted to present reading assignments, questions, and replies to student answers. Student responses are typed on the

keyboard and entered onto the computer for comparison with alternatives previously stored by the author, and the next computer reply is determined by the answer with which a match is established. The teaching stations have been adapted for control of accessory hardware such a random access visual and audio files. The system has the capacity to accumulate and summarize data on student performance for the author.

Even though many of Coursewriter's features are experimental and have only been used in a laboratory environment, it is a good first step toward the development of a system with which a teacher can define a teaching session on a machine. However, Coursewriter does not allow such flexibility as the dynamic creation of problems to be presented based only on student responses. Systems of the future must contain such capabilities.

PLATO (Programmed Logic for Automatic Teaching Operations) is a system developed at the University of Illinois designed to provide an automatic teaching system sufficiently flexible to permit experimental evaluation of a large variety of ideas on automatic instruction. Control over the system is exercised by the ILLIAC computer. Each student communicates with the system by means of a keyset which resembles a typewriter. The system communicates with the students by means of closed-circuit television screens on which may be displayed instructional slides. In addition, each student has at his disposal a storage tube which is used to display all material which cannot be prestored on

slides. Actually, the closed-circuit television and the storage tube are built together so that it is possible for the system to display the student's answer to a question, superimposed on a slide at the appropriate place. The system can accommodate a variety of subjects such as French and mathematics using the same basic program by merely reading a suitable parameter tape into the computer and changing the slides appropriately.

PLATO's stated objective is that of experimentation rather than being a useful educational tool, and the objective is met fairly well. The system is limited to use of the ILLIAC computer and is too expensive for practical general use. The fact that the same basic logic is used for all types of subject matter adds simplicity to the system, but it also introduces limitations which would be unacceptable in the system intended to general use.

One such limitation is the requirement that the student start each teaching session by proceeding through a fixed main sequence of slides, answering correctly each question posed in the course of the sequence. In addition, there is no automatic presentation of subsequences to provide help on a particular concept. Additional material is presented only at the request of the student and the entire subsequence must be executed to completion prior to continuing. Memory limitations of ILLIAC preclude the use of secondary help sequences. Thus, if a student cannot submit a correct answer to help sequence problem and asks for help once more, the machine informs him that no additional help is available.

PLATO keeps records of each move made by a student and the time elapsed at each move. This data is subsequently processed to produce information about the students progress. Care must be exercised in using response time as an evaluation factor since it is not known whether an excessive elapsed time resulted from lack of understanding or lack of attention. Finally, the amount of button-pushing required of the student is excessive and would be particularly troublesome for younger students.

A special-purpose computer designed especially for CAI is the TRW Mentor. The device itself contains the necessary logic for scoring, deciding, selecting, and controlling. Student responses are automatically scored, and decisions are made by the machine on the basis of the scores. The machine selects the subsequent presentation on the basis of the decision and controls the conditions of the presentation. Automatic scoring capability includes single-item tests, multiple-item tests, or a complete test battery. Both the response and the time taken to respond are recorded permanently. Presentation can be synchronized visual and autitory stimuli or either of these alone. Visual presentations can be still frames, motion pictures, drawings, animation, text, charts, or graphs intermixed in any desired order. Auditory stimuli can be speech, music, or any other sounds. Presentations can be time controlled by the machine or can be paced by the student.

The TWR Mentor is a device capable of handling only one student at a time, and therefore suffers from a high cost per student-hour on the machine. If the device were designed as a

multi-student machine, much hardware duplication could be avoided resulting in lower cost. The most serious limitation of Mentor is the lack of dynamic changeable graphics displays. While a wide variety of presentations can be made, all are preprogrammed and entered into the machine on film.

In 1967, Stolurow commented that, at the time, CAI was like the Wright brothers' first airplane: it is hardly of practical value, but its development cannot be ignored. Although there have been some very significant developments, computer-aided instruction has not yet reached its full potential. A complete interactive graphical computer-aided instruction system such as the one proposed later in this thesis has yet to be implemented or even attempted. However, it is felt that it is only a matter of time.

D. THE PROBLEMS OF CAI

1. Hardware Problems

Hardware is probably the least important problem facing the further development of computer-aided instruction systems. In the overall computer industry, hardware is generally more advanced than software. Machines are in existance today which could be used to implement very powerful, effective CAI systems. However, further development of time-shared machines designed specifically for computer-aided instruction will provide CAI system designers with more flexibility. The development of better input-output devices is of particular interest. With the advent of reliable,

economical color graphical display units, the designer will have at his disposal another dimension with which to work.

2. Software Problems

Most of the tools required to write good CAI software are at hand. Basic systems have been written in conventional computational and character manipulation languages, and other specialized CAI languages have been devised which have proven to be useful. However, if CAI is to move on into the second generation, more powerful special purpose languages will be required. The main hurdle to computer-aided instruction software is the set of intrinsic problems discussed below.

3. Social and Economic Problems

Education is a complex system consisting of students, teachers, principals, parents, school boards, taxpayers, and city, county, state, and federal governments all of whom have different and often conflicting ideas on the process of education. One of the biggest obstacles to the rapid and effective introduction of computer-aided instruction into the schools might prove to be the school system itself which has historically been resistant to change. A major institutional change that could encourage experimentation, flexibility, and variety might be required before this new technology can contribute significantly to education.

Economic considerations might also delay the widespread implementation of CAI. Hardware purchase and rental is still expensive, particularly for educational institutions with limited

resources. Also, CAI software development, as with all areas of software development, is a very costly and time-consuming process.

4. <u>Intrinsic Problems</u>

The most difficult problems impeding the progress of computer-aided instruction are those involving the basics of the educational process itself and the inherent technical complexity of CAI from the point of view of the educators. While a good CAI system will not appear complex to the students and teachers who use it, it will be difficult for even knowledgeable educators to evaluate existing and proposed new systems for their applicability and desirability.

Even more basic is the problem of the analysis of subject matter for converting it into instruction sequences. The subject matter must be analysed into units that can provide the building blocks by which the student goes through an instructional program. Once the blocks are established there still remains the problem of sequencing and flow of the blocks so as to present the material in an optimum manner.

Another problem is the use of different forms of response to indicate the attainment of knowledge and understanding. What type of response whould be required of the student for a given educational evolution, and will the response be a true indicator of the student's understanding of the concepts involved? These are difficult questions.

Although some work has been done in the "intrinsic" problem areas of computer-aided instruction, much remains to be

done. Until the many factors involved in the educational process itself are more fully explored and more thoroughly understood it will be difficult for CAI to realize its full potential.

III. A COMPLETE INTERACTIVE GRAPHICAL CAI SYSTEM

A computer-aided instruction system must include certain basic functions in order to meet the definition stated in paragraph I and in order to constitute even a limited tool in the educational process. However, in order to be really effective, a system must go beyond the bare necessities and provide maximum utility to the tacher and at the same time give the student as much diversified interaction as possible.

This section contains a discussion of a proposed complete interactive graphical computer-aided instruction system designed specifically for teaching elementary mathematics at the fourth grade level. Even though the system is specific in nature, the concepts presented are general and could be applied to other areas of instruction. The discussion is divided into two parts. The first part concerns those functions provided by the system which enable the teacher to describe teaching sessions to the machine and to perform certain administrative functions. The second part deals with those functions provided by the system which interact with the student to achieve the educational goals.

The two most important factors in the discussion of an interactive computer-aided instruction system both from the teacher's standpoint and from the students' point of view are first of all how the system interacts with the user and secondly, what educational facilities are available in the system. The interaction must be smooth, fast, reliable, uncomplicated, and human oriented. It

should be assumed that the typical user is totally unfamiliar with computers. He must not be required to perform an excessive amount of button-pushing or knob turning, and explicit instructions must be provided whenever needed. A good CAI system must really prove itself to be a useful tool to the teacher, and not just another gadget. And to the student it must be an interesting, diversified educational experience tailored to his individual needs which will not only supplement this classroom learning, but which will open new horizons and encourage individuality, originality, and creativity. It is felt that the system presented here goes a long way toward meeting these requirements.

A. TEACHER-COMPUTER INTERACTION

Conventional educational materials such as books, films, and some of the earlier CAI systems are prepackaged and sent to a teacher for his use, with the teacher having no freedom to change the material to meet his particular needs. It is often the case that, while portions of the available material are quite adequate, other portions do not meet a specific teacher's or an entire school's needs. The alternatives available to a teacher in such a situation are limited. He may use the inadequate material in spite of its shortcomings; he may discard the deficient material and proceed without instructional aids; he may search for additional material which is more to his liking; or he may rewrite the inadequate sections of the material to fit his needs.

The first alternative is undesirable since the use of educational material not suited to a group of students is a great

injustice to the students. The second alternative, while probably preferable to the first, deprives the students of helpful aids and increases the teacher's burden. Additional material which does meet the specific need might not be available, and a search could prove to be time-consuming. Finally, rewriting material could be impossible, as in the case of films, or impractical in other cases. The time required for a teacher to rewrite a section of a book, if indeed he is capable of doing so at all, is excessive.

The proposed CAI system would be prepared and made available to educators in much the same way as conventional educational materials. Entire state, county, or city school systems might find that the CAI package as received is exactly what is needed and would use it without modification. On the other hand, school A, having a large percentage of students with a certain ethnic background, might feel the need to modify portions of the system to reflect this difference. Also, teacher X, teaching a class of particularly advanced students, might wish to change his version of the system to include more advanced topics, using some teaching techniques which he has developed himself and which are not available in the system as received. These capabilities are available in the proposed system.

1. Primary Functions

The primary functions of the system are those which allow the teacher to interact with the computer to define or modify teaching sessions. The teacher is allowed a great amount of flexibility in the method used to describe the various aspects of a session and in the amount of detail used in specifying session

parameters. Emphasis is placed on graphical interactive presentations which provide maximum human-oriented aid to the teacher.

In starting to develop a new teaching session, a teacher must be given a certain amount of basic preliminary information about the system being used. In the proposed system, this basic explanation can be provided in the form of text, drawings, diagrams, moving pictures, or any combination of these modes. Throughout the session, additional explanation is provided at certain points, and more detailed help is given when called for by the user.

Once the user understands the workings of the system, he may begin to develop his session. A logical place to start is with the first frame or item to be presented to the student. The system must provide the teacher with a convenient way of specifying just what will be presented to the student during execution time. The proposed system displays a picture to the teacher similar to that shown in figure 1. The elipse in the figure represents the initial frame to be presented to the student. Once the teacher has completely described the first frame, the next logical step is to describe the second frame. This can be done in much the same way as the first. The remaining frames in the session can then be described in turn.

Some method is needed, however, to help the teacher to remember what frames have been completed and to relate the frames to the sequence in which they will be presented. One way of accomplishing this is to display the first frame as shown in

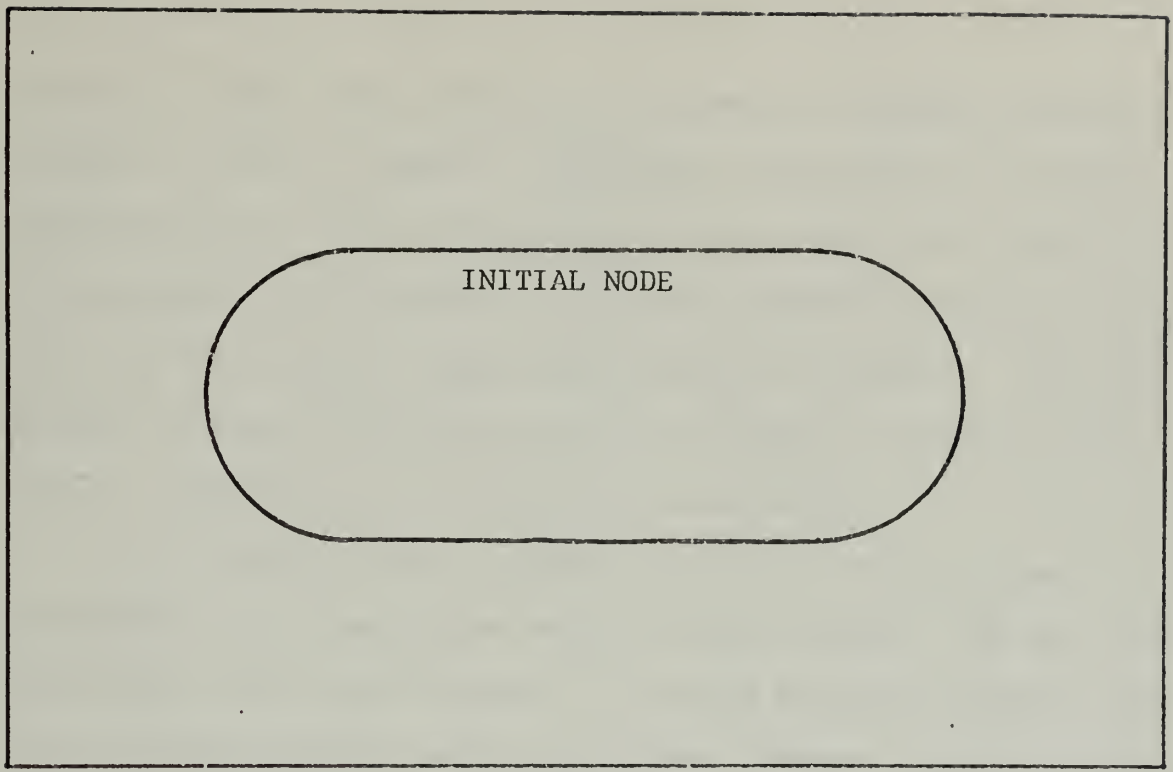

FIGURE 1

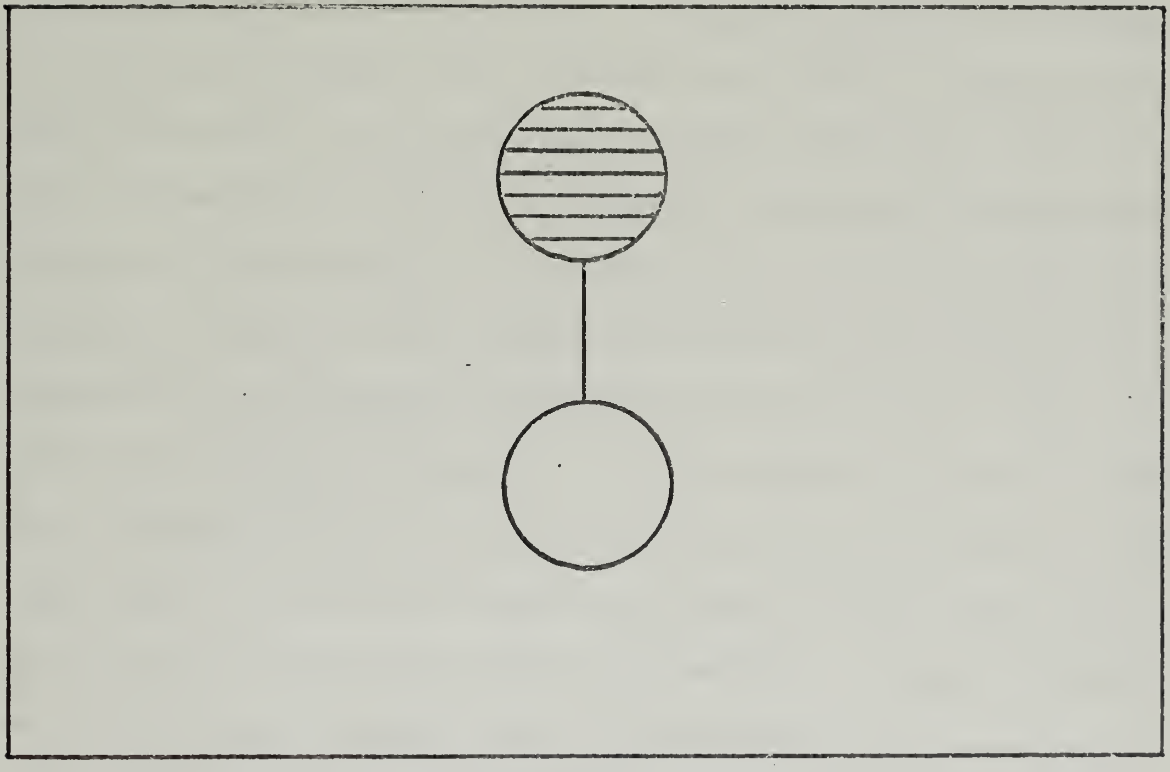

FIGURE 2

figure 1. Then, upon completion of frame one, display a picture similar to that in figure 2, indicating that frame one has been completed and is to be followed by a second frame which has yet to be completed. The teacher would then be allowed to complete frame two with the aid of a large node as shown in figure 1. The completion of frame two would result in a display as shown in figure 3. This process could continue indefinitely.

At the start of a teaching session the type of linear presentation discussed above would be quite common. The same explanation, instruction material, or review material might be presented to all students using the system. However, as the session progresses to more individualized interaction, a need arises for the presentation of material which is dependent upon the responses of the various students, and a teacher must have the capability of specifying what is to be done under certain conditions. The concepts represented by figures 1 through 3 can easily be expanded to accommodate this facility. Upon completion of the third frame, as depicted by figure 4, the teacher might wish the next item to be presented to be dependent upon the student's response. This can be indicated as shown in figure 5. In this example, the fourth frame to be presented to the student will depend upon his response to the third frame. This process may be continued as before until the description of the entire session is completed. Figure 6 depicts a session in a more advanced stage of development.

The structure described above is commonly referred to as a tree. The circles in the tree are called nodes and the lines

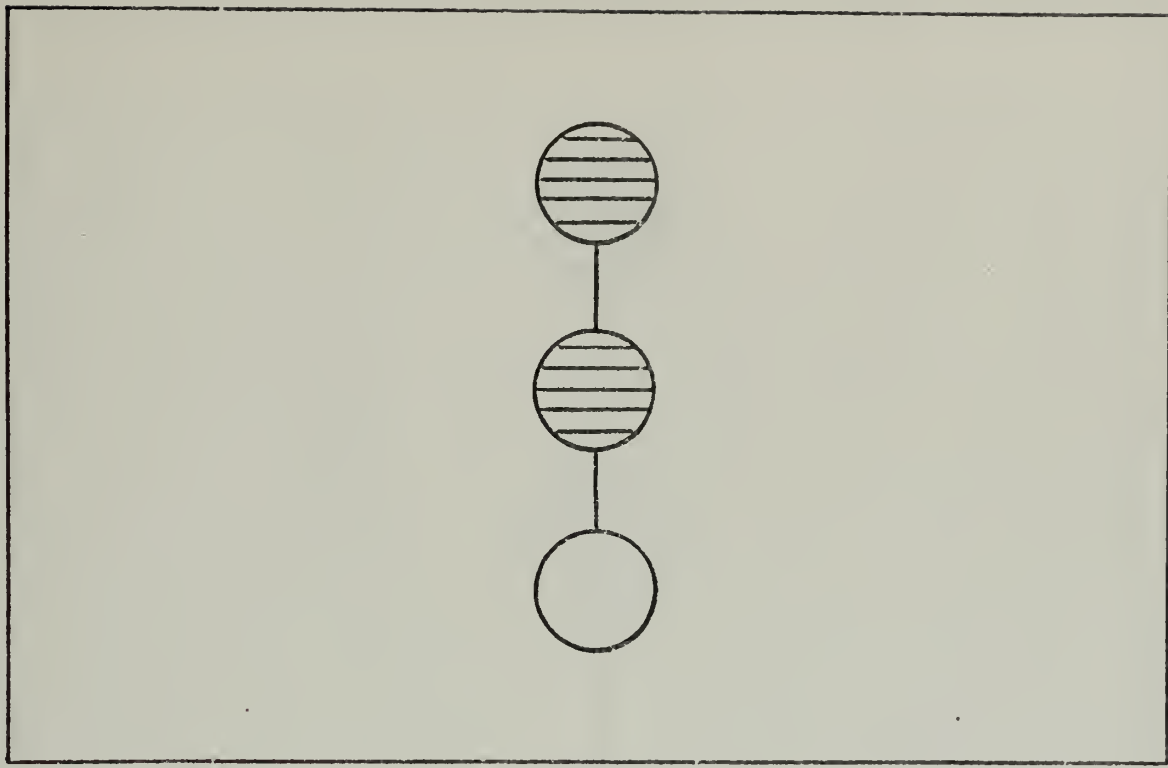

FIGURE 3

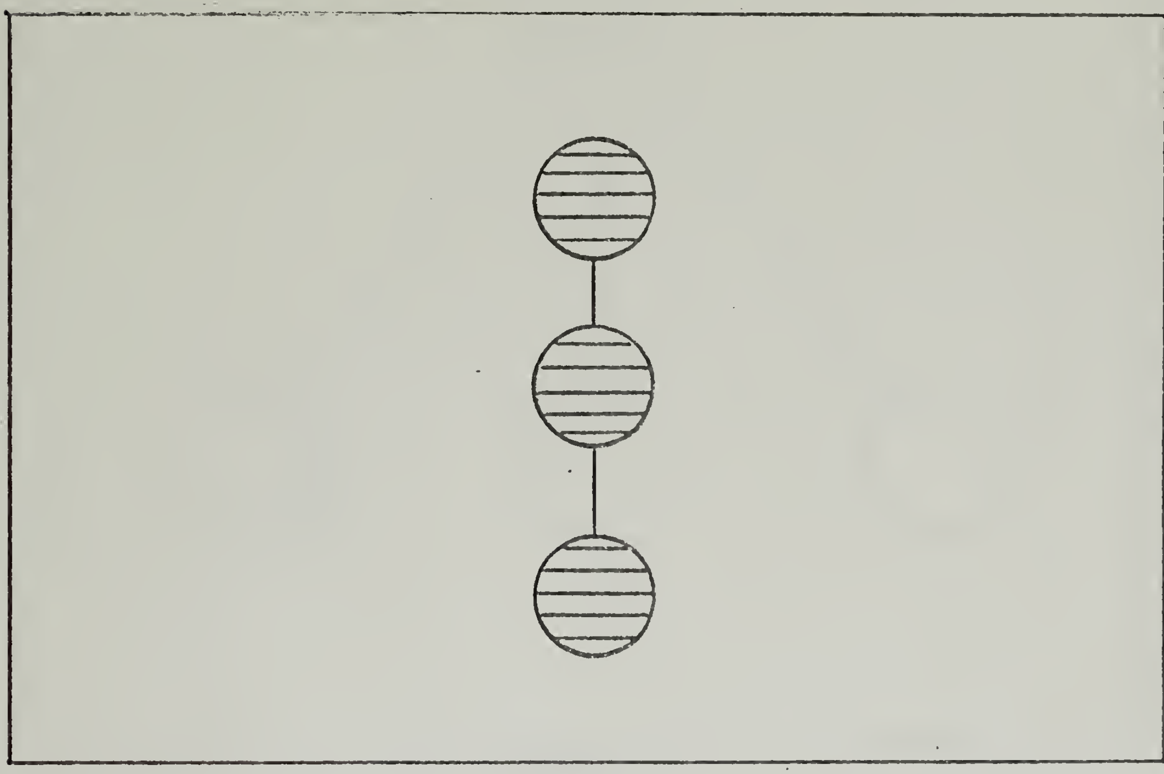

FIGURE 4

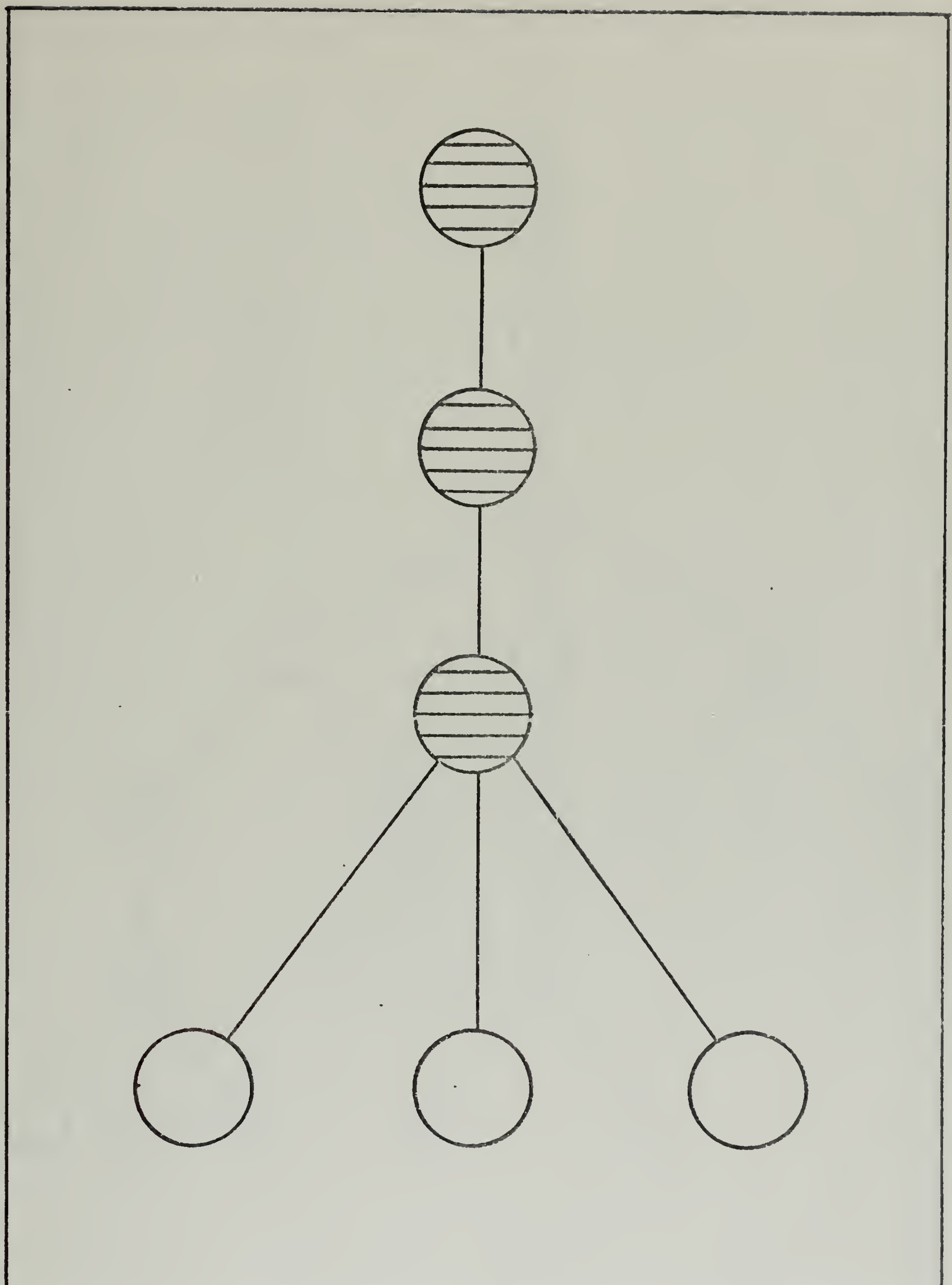

FIGURE 5

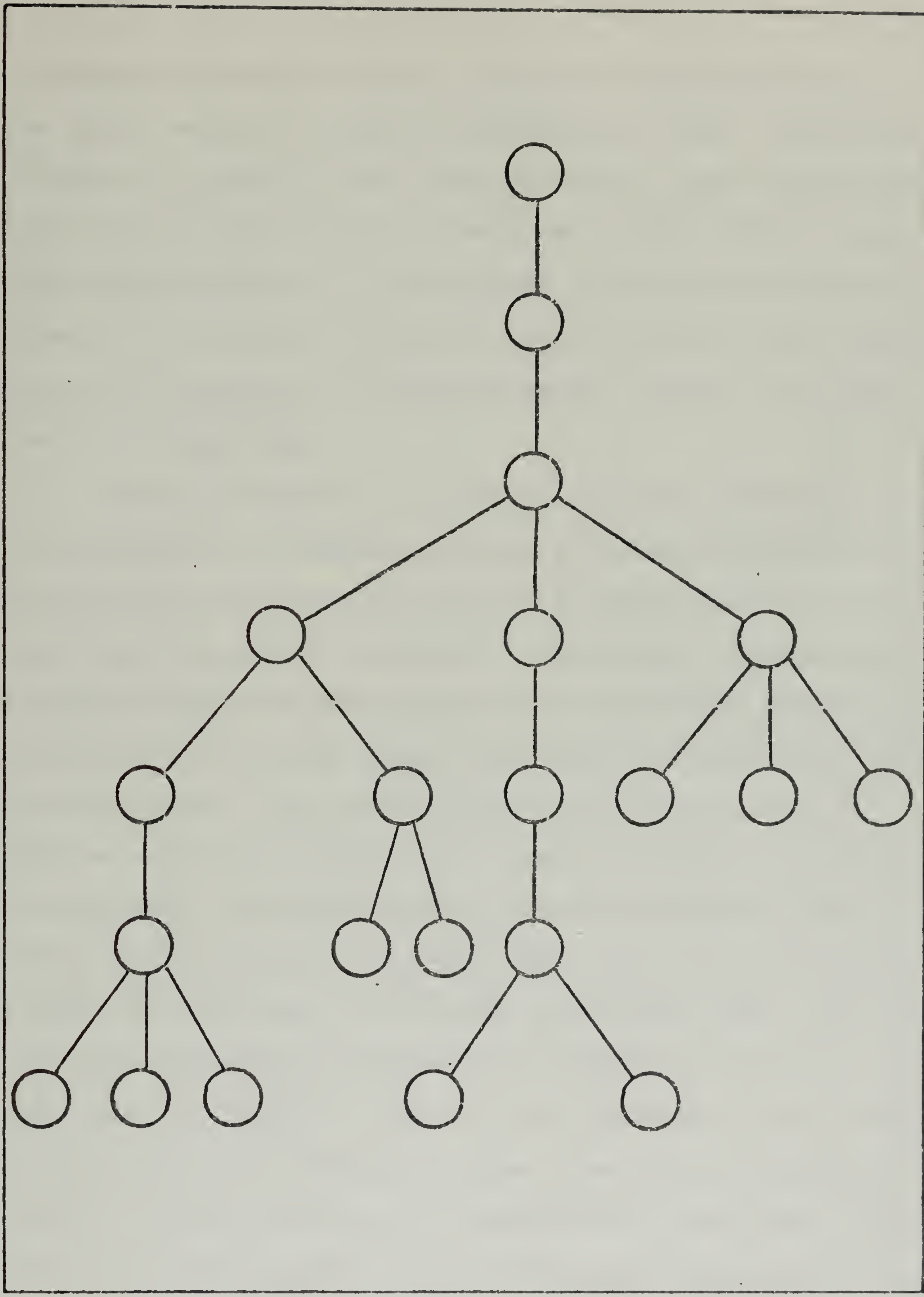

FIGURE 6

connecting the nodes are edges. A subtree is any set of one or more nodes eonneeted direetly or indireetly by edges. No attempt is made here to define these terms rigorously. They are, however, used extensively throughout the remainder of this thesis. The most eommon definition of trees exclude loops such as those contained in the structure in figure 7. Structures with sueh loops are normally referred to as directed graphs. However, the term tree will be used here.

Having established, in general, the basie method used in the development of a teaehing session, it is now appropriate to examine some of the specific functions in greater detail. The first thing that must be available to the teaeher is the ability to describe what is to be presented to a student in a given frame as represented by a node in the tree. There are three basie types of presentations. The first is general text material such as might be used at the beginning of a session to issue instruetions to the student. The seeond type is problem presentation and includes the displaying of arithmetie, set theory, and other types of problems to the student in a variety of different forms. The third type of presentation is tutoring which requires the eomputer to interact with the student in order to help the student grasp concepts whieh he does not yet fully understand. One way of allowing the teaeher to speeify what is to be presented by a given node is the naming of a system routine together with applicable parameters. By naming a routine in a node, the teacher speeifies that the specific routine is to be called during the student-computer teaching session.

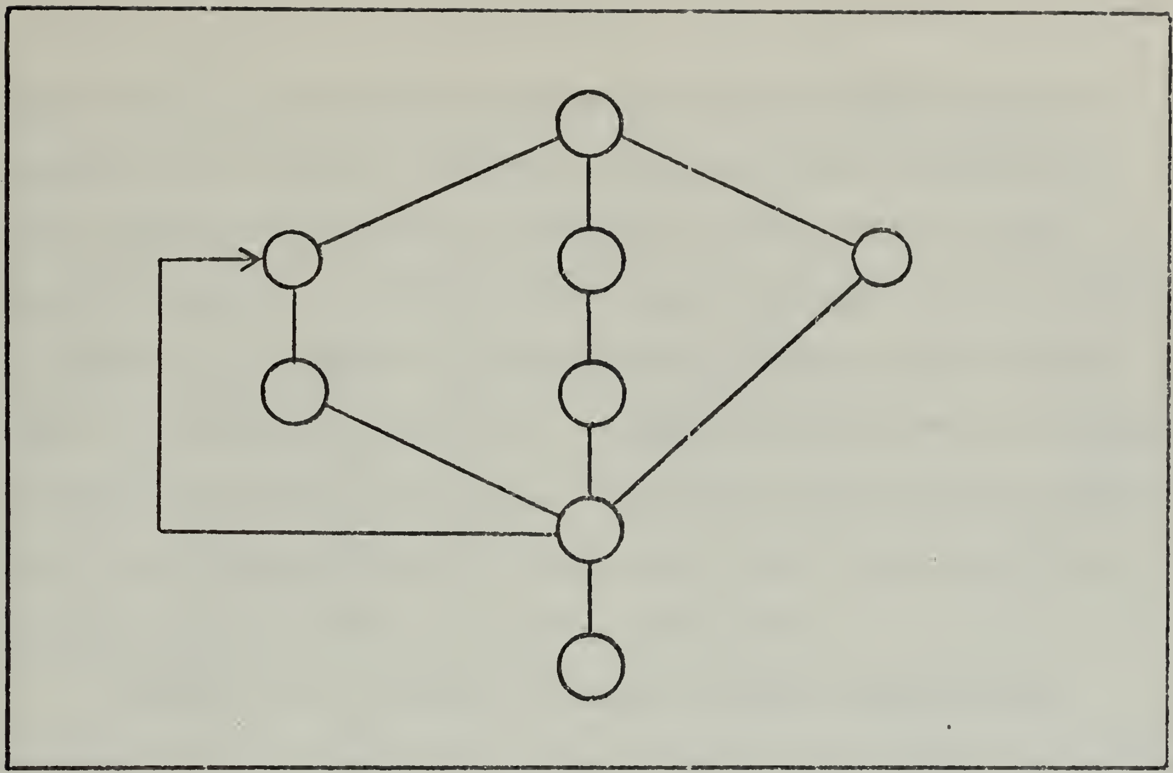

FIGURE 7

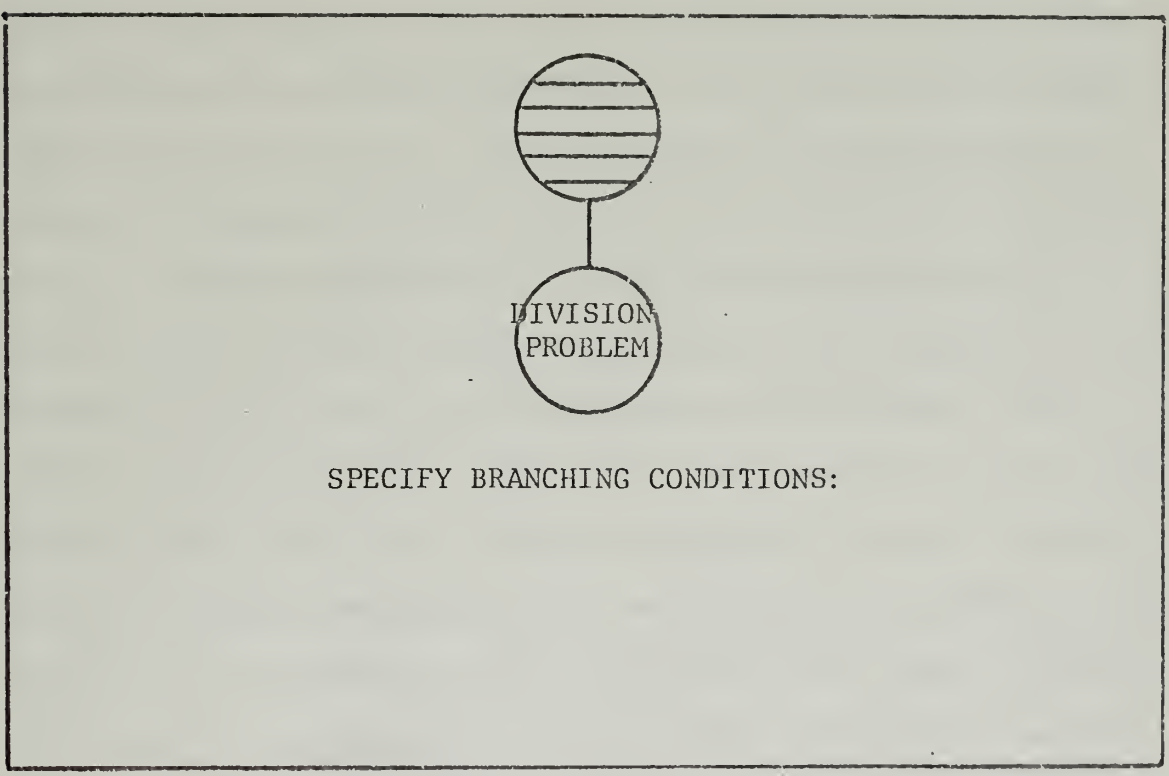

FIGURE 8

These system routines are included in the prepackaged system as received by a teacher, and also the system allows a teacher to write additional routines for inclusion in his version of the system if desired. System routines might provide for such things as displaying an arithmetic problem and allowing the student to solve it directly on the screen, solving a problem, which a student has previously done incorrectly, in motion picture fashion indicating to the student what errors were made, and illustrating the concept of associativity using the number line.

Another way to define a node, and thus specify what material is to be presented to the student, is by using the general horizontal mode. It is limited to the definition of arithmetic and set theory problems which are to be presented to the student who performs the indicated operations. An example of this mode might be (8 + 23) - 14 =. In this case the actual problem as written is displayed on the screen and the student enters the answer 17. If the specification of the problem were given as $(X_1 + Y_1\ Y_2) - Y_1$, the system would assign random integers to the variables X_1, Y_1, and Y_2 and the problem that appeared to the student might be (9 + 82) - 8 =. This is a powerful tool for the teacher since he may use a similar expression in many different places in a given teaching session and, in general, different integers will be assigned to the variables by the system. In this way a teacher may emphasize a given concept throughout a session without taking the time and effort to devise specific problems. He need only specify the class of problems to be presented.

However, if desired, the system can present to the student a specific problem with which the student has had trouble in the past. Fixed integers and variables may be mixed in a given problem description if desired and the system will display the fixed integers and assign appropriate integers to the variable. Any of the four arithmetic operators together with any number of variables or fixed integers and sets of parentheses may be used.

The system examines the operators to determine whether the problem being defined is arithmetic or set theoretic. In set theory problems, sets must be named by single letters such as A, B, and C and operators such as intersection and union are allowed. Graphical representations of the named sets can be displayed to clarify the problem for the student.

Another way to define a node, which is very similar to the general horizontal mode in the general vertical mode. The only difference in the two modes in that in the vertical mode, problems are displayed for the student in normal hand-calculation form; therefore, a specification of

$$\begin{array}{rr} X_1 & X_2 \\ Y_1 & 3 \\ +\ 8 & Z_2 \\ \hline \end{array}$$

might result in the following problem being displayed to the student:

$$\begin{array}{r} 57 \\ 13 \\ \underline{+\ 82} \end{array}$$

In both the general horizontal and general vertical modes, additional parameters may be specified in several ways. The first way is a semi-freeform series of phrases such as "no number greater than 25 or less than 10," "problem to include carry," and "provide for a non-zero remainder." The system scans the phrases for key words and takes the necessary action. The second way of specifying additional parameters is by using a more restrictive but more compact format. Examples of this type are: "$X_1 > X_2$," "$X_2 + Y_2 > 9$," "$X_1 X_2 + Y_1 Y_2 = 0$," and $Y_1 - X_1 > 0$." As many restrictions may be placed on a problem as desired and either of the above forms may be used separately or intermixed.

Using the type of system routines described above, together with an additionally implemented routines, a teacher may easily assign meaning to the nodes in the session tree thus specifying what is to be presented to the student. Having established this, some means is needed to specify the conditions under which a branch of the tree is to be taken. This is done by allowing the teacher to enter any number of conditions upon which the branch depends. If no conditions are entered, then the system assumes that the current node is followed by a single node which is to be branched to unconditionally. If two or more conditions are entered, the system provides for the establishment of the proper number of succeeding nodes and sets up the appropriate checking mechanism to determine which branch is to be taken.

The teacher can use the characteristics of the problem as described as an aid in determining what branching conditions to

specify. For example, if a "carry" is to be included in an addition problem, it would probably be advisable to include a carry check together with appropriate branches. Also, the language used to specify branching conditions is the same as that used to define nodes, thus relieving the user of the requirement to learn additional languages.

The branching conditions specified by the teacher are used as labels for the edges in the tree so that the teacher will know at all times which node is to follow any given node and under what circumstances. An example of the specification of branching conditions is shown in figures 8 through 10. In figure 8 the teacher has completed the specification of a division problem and wishes to have the next node dependent upon the student's response. If the student's answer is correct, one branch is to be taken. If the quotient entered by the student is incorrect, a second branch is to be taken, and if the remainder is incorrect, a third branch is to be taken. In figure 9, the branching conditions have been entered by the teacher. The system determines that three nodes are required to succeed the division problem node. Figure 10 shows the resultant picture displayed to the teacher, with the three successor nodes added and the corresponding edges labeled appropirately.

Some mechanism is needed to enable the teacher to indicate to the system which node the teacher wishes to work with next. At a point in the development of a session such as is shown in figure 10, the teacher may wish to handle any one of the three nodes immediately and leave the other two until later. Upon

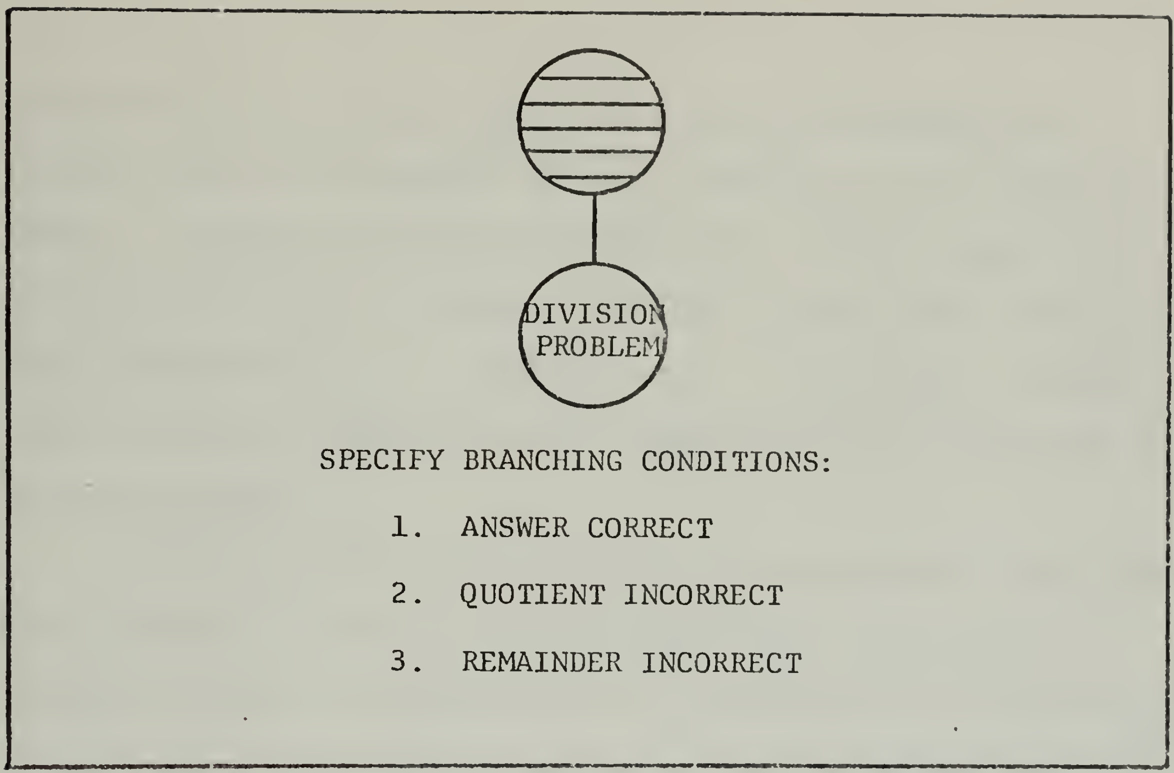

FIGURE 9

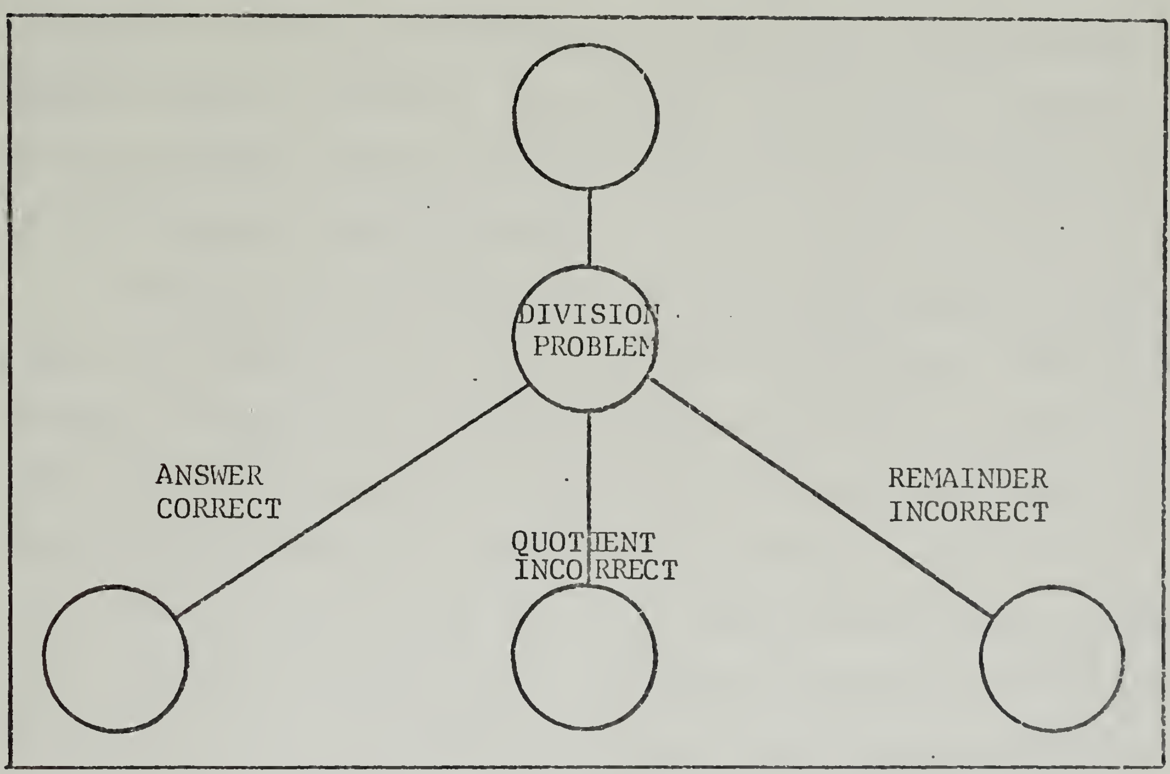

FIGURE 10

completion of the chosen node, it may even be desirable to continue with its successors prior to dealing with the other two nodes. This node selection facility can be easily handled in the proposed system by allowing the user to touch with a light pen that node which is to be completed next. In this way, a teacher may complete the definition of the nodes in the tree structure in any desired order.

As the process of describing a teaching session using the tree structure progresses, it is apparent that a point may be reached at which no additional space is available on the graphical display device. When this situation occurs, the system automatically scales the entire tree to make more space available. However, the amount of scaling that can be done while still maintaining a legible display is limited. As the nodes and edges are scaled, the corresponding labels are abbreviated and truncated so as to fit into the smaller space available.

Another useful system function allows the teacher to display only a portion of the tree structure on the screen. When this is done, and more space becomes available as a result of fewer nodes, edges, and labels being displayed, the system automatically reverts back to the most complete labels possible. Thus if the entire tree has been scaled to provide additonal space, and a label in its abbreviated form is not sufficiently descriptive, the user may display only the applicable portion of the tree in order to be provided with a better label. The teacher may again display the entire tree or any other portion of the tree at will. This

partitioning feature also allows the user to work with one part of the session without having the screen become too filled with material in which he is not immediately interested. An example of this process is illustrated in figures 11 and 12. Only that portion of the tree in figure 11 which is enclosed in the dotted line is shown in figure 12. Notice that the abbreviations used in labels in figure 11 are no longer needed in figure 12.

At any time during the development of a session the teacher may wish to add to, delete from, or revise portions of the existing tree structure. He may even wish to delete the entire tree and start over from the beginning. This type of editing may easily be done on the proposed system. Individual nodes or groups of nodes may be added at any point in the tree by the use of an insert function. A delete command causes one or more nodes to be deleted from the structure. Changes may be made to any existing node or edge or their corresponding labels by using a modification function.

Using the tree structure to define the overall flow of the session, the branching conditions to indicate which path is to be taken, the node definition facilities to specify the exact material to be presented, and the modification functions to change the session as necessary, a teacher may create a new teaching session. This creation process is made easier for the teacher and the end result is more effective for the student as a result of the interactive graphical facilities provided to the teacher by the system. In addition to the creation of new teaching sessions, the system can be used for another very important purpose. If a prepackaged

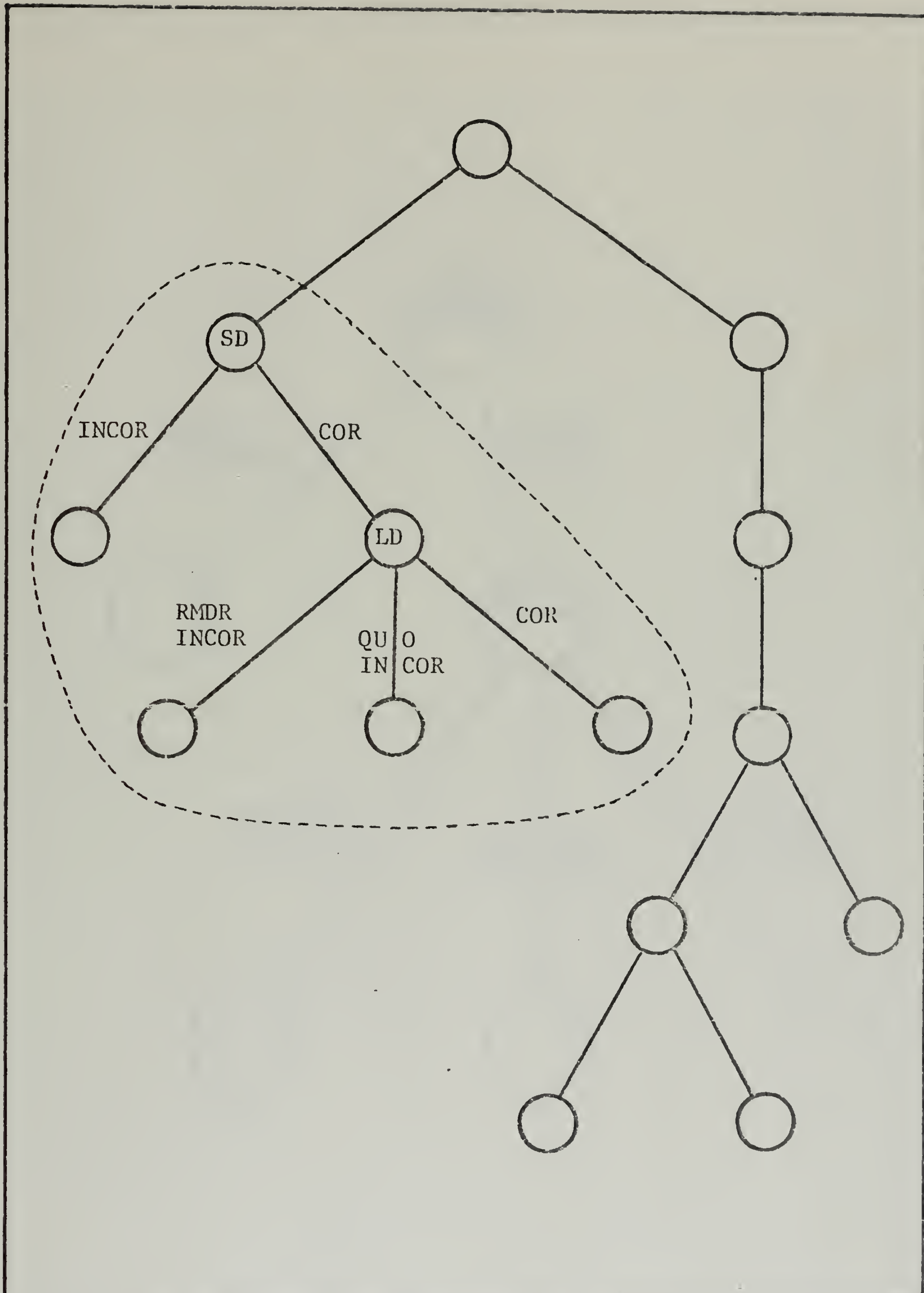

FIGURE 11

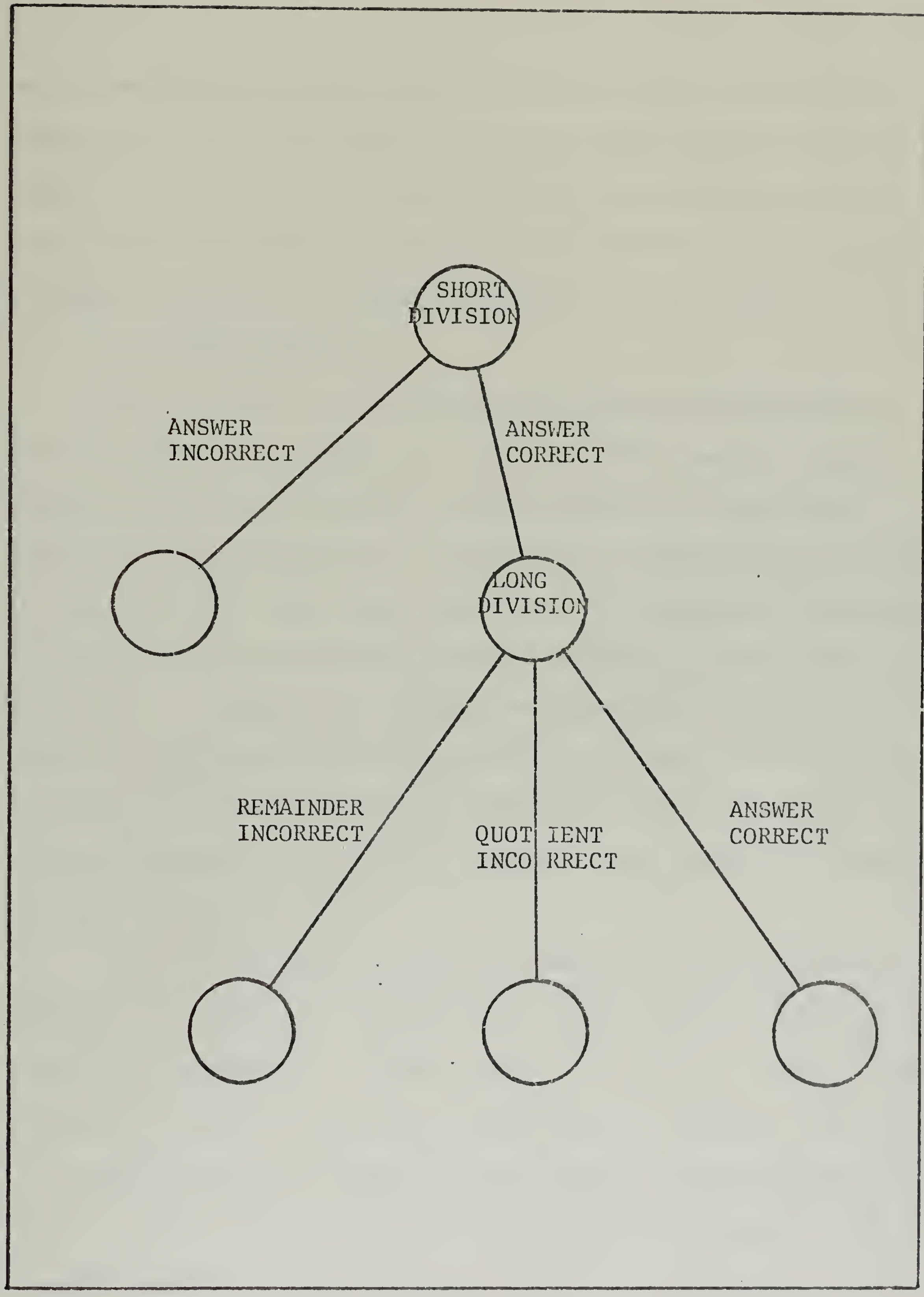
SHORT DIVISION
ANSWER INCORRECT
ANSWER CORRECT
LONG DIVISION
REMAINDER INCORRECT
QUOTIENT INCORRECT
ANSWER CORRECT

FIGURE 12

set of sessions generally meets a teacher's needs, but requires modification in certain areas, the same system functions used to create new sessions may be used to modify the existing sessions. This allows the teacher to tailor a set of existing sessions to his exact requirements with a minimum of time and effort.

2. <u>Secondary Functions</u>

The secondary functions provided to the teacher by the system include those functions which allow the teacher to input general parameters concerning a given session or a particular student and to perform other functions of an administrative nature. For instance the teacher may specify one or more general parameters to the system concerning the overall educational level of the entire class, including the inclusion or omission of any type of problem which has or has not been covered in class. The teacher is also able to input parameters concerning specific students' general advancement level and achievement level in specific areas of a given session.

Another secondary function provided by the system is the ability of the teacher to specify the type of feedback desired at the end of each session. He may receive a detailed listing of the students' responses to each item presented, or a general indication of how each student performed in each distinct area presented, or he may receive only the percent of total correct responses given by each student.

A third secondary function, and perhaps the most useful, is the ability of the teacher to access the files kept by the

computer on each student and have displayed on the screen a summary of a particular student's progress. This function would be particularly useful as a guide in counciling the student or in conducting an interview with a student's parents.

B. STUDENT-COMPUTER INTERACTION

The type of material that is presented to a given student during a teaching session, as well as the manner in which it is presented, is determined to a large extent by the teacher who develops the session. This section contains a discussion of some of the facilities which may be specified by the teacher for use with students and some functions which are available to the student as a built-in part of the system. The discussion is divided into two parts; primary functions and secondary functions.

1. Primary Functions

The primary functions include those capabilities of the system which actually do the teaching. Of prime importance in any educational process is the obtaining of the student's attention and the maintaining of his attention throughout the session. The proposed CAI system has an advantage from the start in that the current generation of students are used to watching television and motion pictures and seem to be more inclined to be attentive to these devices than to adult humans. The system provides for a great variety of presentation methods including text explanations, pictures, graphs, diagrams, and so forth. Motion, color, and varied intensity can be used to great advantage in varying the presentation and maintaining the students' attention.

The first function to be considered is the presentation of new material to the student. While some teachers may prefer to present new material in the classroom and use the CAI system for review and drill and practice, others may wish to have the computer restate the presentation in a different form to strengthen the students' understanding or he may wish to leave the job completely up to the system and use classroom time for review and the answering of questions. In any case, the presentation of new material is an area in which the system can use a great amount of variation and do a very good job.

In addition to the presentation of new material, the system allows for periodic review of material previously covered. The amount of review material presented is determined by the teacher in his development of a session, and the system also injects review material based upon the students' understanding or lack of understanding of specific concepts.

Drill and practice sessions are another important function of the system. The system can present such sessions to the student immediately following review sessions or the presentation of new material to allow the student to apply the principles involved, and can then check his responses for accuracy. Incorrect responses can be followed by a tutoring session including showing the student what his mistake was and illustrating the correct way of solving the problem. A detailed analysis is made on all incorrect responses as well as any intermediate results inputted by the student. This analysis is a very important feature of the system because it

allows the system to determine a student's specific area of deficiency and to tailor further action appropriately. The student is also allowed to defend himself on points that he feels are correct. From this interaction the system can gain further insight into the reason for a student's confusion on a particular concept.

The student is rewarded for correct responses by the self-satisfaction of seeing that his answer is correct. The system further rewards him with encouraging messages upon receiving a correct response and by informing him when he has reached certain stages of attainment in the session. Each student is encouraged to exhibit individuality, innovation, and creativity by allowing him to draw pictures and diagrams and to construct some of his own problems to solve. Throughout all teaching sessions, whenever possible, pictures, diagrams, and motion pictures are used instead of words to convey ideas. All graphical displays presented to a student are geared to his level of understanding. A fourth grade student is shown more simplified pictures, drawings, and diagrams than a seventh grade student.

The prepackaged sessions are designed carefully to use text which includes vocablulary appropriate to the students who will be using the sessions. Teachers must also be careful in creating their own sessions to tailor the English used in test presentations to the students' level

2. Secondary Functions

The secondary functions provided by the system to the student are of an administrative nature. A complete, detailed history of each student's progress and status for the current teaching session is maintained by the system, including all responses made and an updated evaluation of the student's current achievement level. This information is not seen by the student but can be obtained by the teacher. A more general, perminent, long-term history of each student's educational status and achievement levels for each of the various concepts and general areas of instruction is also maintained. This is also available only to the teacher.

The student has the capability of inputting messages which are printed directly on the teacher's report of the session. In this way the student can make comments and ask questions which the teacher can resolve during the next class period or individually as appropriate. The student may also call a proctor in case of hardware failure or other problems he cannot handle.

IV. IMPLEMENTATION

The proposed system was implemented in XDS FORTRAN on an ADAGE Graphics Terminal model 10 (AGT-10) interfaced with an XDS 9300 computer, and consists of a series of subroutines which perform the various system functions. This implementation concentrates on the student-computer interative portion of the system. It currently has the capability to teach aspects of fourth grade mathematics by doing such things as presenting addition, subtraction, multiplication, and division problems to the and allowing the student to solve the problems directly on the screen. The answers can be checked and the student informed as to their accuracy. Various routines are available to present new and review material to the student using varied graphics techniques. Finally, a routine can be used to display an arithmetic problem to the student in motion picture fashion, working the problem step by step in the same way that a student would work it.

As an example of the implemented system's capabilities, consider the following addition problem. The teacher's specification of the problem was:

$$\begin{array}{r} X_1\ X_2 \\ +\ Y_1\ Y_2 \\ \hline \end{array}$$

To ensure that the problem would include a carry the teacher also specified $X_2 + Y_2 > 9$. The teacher indicated that if the student responded incorrectly to the problem, the next presentation was to be a motion picture which would solve the exact same problem

correctly for the student. Thus, during the execution of the session, the system assigned random digits to the variables, making sure that the sum of the two units digits exceeded nine, and displayed the following:

```
  24
+ 57
----
```

The student correctly entered a one in the units position of the answer; then, forgetting to carry, entered a seven in the tens position. The final display was therefore:

```
  24
+ 57
----
  71
```

Upon checking the student's response and analysing the reason for his error, the system, having been told to do so by the teacher, informed the student that his answer was incorrect and proceeded to display the correct solution to the problem as follows.

The original unsolved problem was again displayed on the screen together with text telling the student to watch each step closely so as to understand his mistake. After giving the student sufficient time to read the text, the system proceeded with the solving of the problem in a step-by-step, motion picture fashion. First, a one was displayed in the units position in the answer:

```
  24
+ 57
----
   1
```

After a delay of about five seconds, the one carry was displayed above the problem in the tens position:

```
  1
  24
+ 57
----
   1
```

The carried one that appeared on the screen was half the size of the other numbers. After another five-second delay, the system completed the answer with an eight in the tens position:

$$\begin{array}{r} {}^{1}\\ 24 \\ +\ \underline{57} \\ 81 \end{array}$$

A final small amount of text was then displayed at the bottom of the screen telling the student to look at the problem again to make sure he understood his error, and to enter a "yes" is he felt that he understood, and a "no" otherwise. The system would provide additional help if a negative response were received and would otherwise either check the student's understanding with another problem or continue with other material.

The routine described above is called DEMO, and the complete listing is included in this thesis. The time delay used in the routine is variable and can be easily increased for slower students and decreased for more advance students. DEMO could, with little programming effort, be made to further emphasize a students specific mistake by circling it or by displaying it with greater intensity. Future extensions along this line might include using a different color to display the critical portion of the problem.

The main advantage of the DEMO routine is that the exact problem that was done incorrectly by a student can be solved for him correctly and his errors pointed out clearly. This facility is available whether the problem was originated specifically by the teacher during the session definition, by the system during execution of the session, or by the student himself during a

creative problem session. Some of the existing systems allow the presentation of a problem of the same general type wolved incorrectly by a student, but few provide for the presentation of the exact same problem.

Possible extentions of the DEMO routine might concentrate more on the tutoring aspects of the presentation. The student might be allowed to stop the display at any point to study it more carefully and to ask questions of the system on points of confusion. The system could either provide simple answers to the questions or provide more detailed graphical displays illustrating the specific concept in question. The system could thus provide many of the advantages of a personal, human tutor with the added quality of infinite patience.

Currently, little has been done toward implementing the teacher-computer interactive portion of the proposed system. However, it is felt that unreasonable effort would not be required to do so. The routines that have been completed with interact with the student have been programmed generally enough so that they could be easily expanded.

V. CONCLUDING REMARKS

The proposed complete interactive graphical computer-aided instruction system is considered to be a useful educational tool which could be used to advantage today. The system provides the teacher with prepackaged sessions which are designed to teach the student at his own particular level using techniques which prove themselves to be most effective for the individual student. Facilities are available for the teacher to modify the existing sessions to meet his needs more effectively with little knowledge of computers and a minimum of time and effort required. The teacher may also use the system to design his own teaching sessions, thus giving the students the benefit of his experience and personal teaching techniques developed through the years. Whatever the use to which the teacher puts the system, he is given the advantage of seeing his work protrayed graphically and dynamically so that he has a better feel for what he has done, what he is currently doing, and what part of the work is yet to be done.

To the student the system represents an enjoyable break in his school routine. The mere fact that he is doing something different for a certain period during the day adds to a student's interest in the system and makes him more receptive to the system as well as to the classroom environment upon his return.

The current stage of implementation of the system is not such that the system could be put to immediate general use. However, the concepts shown in the completed portions of the system are

easily extendable into the work that remains to be done. There are also other considerations which should be taken into account in putting the system to general use. Storage requirements for the completely implemented system would be considerable, nedessitating the use of some sort of direct access storage devices for the storage of the various system routines and for the maintenance of the required records. Routines could be called into core storage only as needed by a user. As a practical matter, the system would have to be implemented a a time-shared computer in order to provide a sufficient number of teaching stations and to reduce the cost per student-hour of instruction.

The proposed system recognizes the individual differences in students and acts as a personalized tutor by determining what each student's educational needs are and then proceeding to satisfy those needs. Full implementation of the system could be accomplished with current hardware and programming techniques. With the improved hardware and software likely to be available in the near future, a system such as the one proposed could be a valuable addition to the field of education.

```
C
C          SUBRØUTINE TØ DISPLAY ADDITIØN PRØBLEMS ØN THE CRT.
C          PRØBLEMS ARE DISPLAYED IN NØRMAL HAND-CALCULATIØN
C          FØRM AND ARE SUBSEQUENTLY SØLVED BY THE STUDENT DIRECTLY
C          ØN THE CRT BY THE SUBRØUTINE SØLVAS
C
C
      SUBRØUTINE ADD
      INTEGER PRØB,NUMS,RANGE,SØRL,REMAIN,CB,ANSWER
      GLØBAL PRØB,NUMS,RANGE,CB,KIGITS(10),SØRL,REMAIN,NUMBER(10),INTERM
     C(10),IDEV,ANSWER,ITDIR(50),IGDIR(50),KNTT,KNTG
C
C     ----- CHECK FØR LESS THAN TWØ NUMBERS TØ ADD -----
      IF(NUMS.GE.2) GØTØ 100
      ØUTPUT(101) 'LESS THAN TWØ NUMBERS TØ ADD ',NUMS
      RETURN
C
C     ----- CHECK FØR IMPRØPER NUMBER ØF DIGITS IN EACH NUMBER -----
  100 DØ 200 I=1,NUMS
      IF(KIGITS(I).GT.0.AND.KIGITS(I).LT.11) GØTØ 200
      ØUTPUT(101) 'INVALID NUMBER ØF DIGITS', I
      RETURN
  200 CØNTINUE
C
C     ----- BØRRØW INDICATØR MUST BE ZERØ ØR ØNE -----
      IF(CB.EQ.0.ØR.CB.EQ.1) GØTØ 300
      ØUTPUT(101) 'INVALID CARRY INDICATØR', CB
      RETURN
C
C     ----- INITIALIZE LØCAL VARIABLES -----
  300 LN=15
      ICP=50
      WNØ=6.
      DØ 400 I=1,10
      NUMBER(I)=0
```

```
  400 CONTINUE
C
C     ----- THIS LOOP HANDLES EACH COMPLETE NUMBER TO BE ADDED -----
      DO 700 I=1,NUMS
C
C     ----- THIS LOOP HANDLES THE SEPARATE DIGITS WITHIN EACH NUMBER ---
      DO 600 J=1,KIGITS(I)
  450 N=RANDIG(WNO)
      IF(J.EQ.KIGITS(I).AND.N.EQ.0) GOTO 450
      NUMBER(I)=NUMBER(I)+N*10**(J-1)
      ENCODE(4,500,ITXT) N
  500 FORMAT(I4)
C
C     ----- DISPLAY ONE DIGIT -----
      CALL TEXTO(IDEV,ITXT,1,LN,ICP,2,1,IER)
      KNTT=KNTT+1
      IF(IER.NE.0) OUTPUT(101) IER,'ADD1',I,J
      ICP=ICP-2
  600 CONTINUE
      LN=LN+1
      ICP=50
  700 CONTINUE
      K=0
C
C     ----- FIND THE LARGEST NUMBER OF DIGITS -----
      DO 800 I=1,NUMS
      IF(KIGITS(I).GT.K) K=KIGITS(I)
  800 CONTINUE
      ENCODE(4,1000,ITXT)
      DO 900 I=1,K
C
C     ----- DISPLAY LINE UNDER THE NUMBERS -----
      CALL TEXTO(IDEV,ITXT,1,LN,ICP,2,1,IER)
      KNTT=KNTT+1
      IF(IER.NE.0) OUTPUT(101) 'ADD2',IER
```

```
      ICP=ICP-1
  900 CONTINUE
 1000 FØRMAT('----')
      ENCØDE(4,1100,ITXT)
 1100 FØRMAT('    +')
      LN=LN-1
      ICP=50-K-5
C
C     ----- DISPLAY A PLUS SIGN -----
      CALL TEXTØ(IDEV,ITXT,1,LN,ICP,2,1,IER)
      KNTT=KNTT+1
      IF(IER.NE.0) ØUTPUT(101) 'ADD3',IER
      RETURN
      END
```

```
C
C         SUBROUTINE TO ALLOW THE USER TO SOLVE ADDITION AND
C         SUBTRACTION PROBLEMS DIRECTLY ON THE CRT IN HAND-
C         CALCULATION FORM.  CARRYS AND BORROWS CAN BE INDICATED
C         ON THE CRT IF DESIRED.
C
C
      SUBROUTINE SOLVAS
      INTEGER PROB,NUMS,RANGE,SORL,REMAIN,CB,ANSWER,STAR1,STAR2
      GLOBAL PROB,NUMS,RANGE,CB,KIGITS(10),SORL,REMAIN,NUMBER(10),INTERM
     C(10),IDEV,ANSWER,ITDIR(50),IGDIR(50),KNTT,KNTG
C
C     ----- INITIALIZE LOCAL VARIABLES -----
      LN=NUMS+16
      ICP=56
      NULL=77777777B
      IBLNK=60606060B
      STAR1=60606054B
      STAR2=54606060B
      ITXT=NULL
      ISTR=0
      IXT=0
      ANSWER=0
      J=1
      K=0
      KN=0
C
C     ----- FIND MAX AND TOTAL DIGITS -----
      DO 100 I=1,NUMS
      IF(KIGITS(I).GT.KN) KN=KIGITS(I)
      K=K+KIGITS(I)
  100 CONTINUE
C
C     ----- INPUT ONE DIGIT OF ANSWER -----
  200 CALL TEXTR(IDEV,NULL,1,LN,ICP,2,1,IER)
```

```
      IF(ISTR.EQ.IXT) KNTT=KNTT+1
      IF(IER.NE.0) ØUTPUT(101) IER, 'SØLVAS 1'
  300 IF(MØD(ITDIR(KNTT),8).EQ.0) GØTØ 300
      ISTR=IXT
  350 CALL TEXTI(IDEV,KTXT,1,0,KNTT,IER)
      IF(IER.NE.0) ØUTPUT(101) IER, 'SØLVAS 2'
      IF(KTXT.EQ.IBLNK) RETURN
      IF(J.GT.(KN+5)) RETURN
      IF(KTXT.EQ.STAR1.ØR.KTXT.EQ.STAR2) GØTØ 400
      DECØDE(4,310,KTXT) ITXT
  310 FØRMAT(I4)
      ITXT=ITXT/1000
      IF(ITXT.LT.0.ØR.ITXT.GT.9) GØTØ 400
C
C     ----- RECØRD ØNE DIGIT ØF ANSWER -----
      ANSWER=ANSWER+ITXT*10**(J-1)
      ICP=ICP-2
      J=J+1
      GØTØ 200
C
C     ----- HANDLE CARRYS AND BØRRØWS -----
  400 ISTR=ISTR+1
      IF(PRØB.EQ.2) GØTØ 500
      CALL TEXTØ(IDEV,NULL,1,LN,ICP,2,1,IER)
      IF(IER.NE.0) ØUTPUT(101) IER, 'SØLVAS 3'
C
C     ----- INPUT CARRY -----
      CALL TEXTR(IDEV,NULL,1,(LN-NUMS-2),ICP,1,1,IER)
      IF(IER.NE.0) ØUTPUT(101) IER, 'SØLVAS 5'
      KNTT=KNTT+1
  420 IF(MØD(ITDIR(KNTT),8).EQ.0) GØTØ 420
      KNTT=KNTT-1
      GØTØ 200
C
C     ----- INPUT BØRRØW -----
```

```
500 CALL TEXTØ(IDEV,NULL,1,LN,ICP,2,1,IER)
    IF(IER.NE.0) ØUTPUT(101) IER, 'SØLVAS 6'
    IF(LN.EQ.(NUMS+16).AND.ICP.EQ.56) GØTØ 520
    ENCØDE(4,510,ITXT) ANSWER
510 FØRMAT(I4)
    CALL TEXTØ(IDEV,ITXT,1,LN,(ICP+2),2,1,IER)
    IF(IER.NE.0) ØUTPUT(101) IER, 'SØLVAS 7'
520 CALL TEXTR(IDEV,NULL,1,(LN-NUMS-2),(ICP-2*KN),1,1,IER)
    IF(IER.NE.0) ØUTPUT(101) IER, 'SØLVAS 8'
    KNTT=KNTT+1
530 IF(MØD(ITDIR(KNTT),8).EQ.0) GØTØ 530
    KNTT=KNTT-1
    GØTØ 200
    END
```

```
C
C           SUBROUTINE TO CHECK USERS RESPONSE TO ADDITION,
C           SUBTRACTION, MULTIPLICATION, AND DIVISION PROBLEMS
C           AND INDICATE TO THE USER WHETHER HIS ANSWER IS
C           CORRECT OR INCORRECT.
C
C
      SUBROUTINE CHECK
      INTEGER PROB,NUMS,RANGE,SORL,REMAIN,CB,ANSWER
      GLOBAL PROB,NUMS,RANGE,CB,KIGITS(10),SORL,REMAIN,NUMBER(10),INTERM
     C(10),IDEV,ANSWER,ITDIR(50),IGDIR(50),KNTT,KNTG
      DIMENSION ITXT(7),NIL(4)
C
C     ----- BRANCH TO APPROPRIATE PROBLEM TYPE -----
      IF(PROB.EQ.2) GOTO 200
      IF(PROB.EQ.3) GOTO 300
      IF(PROB.EQ.4) GOTO 400
C
C     ----- ADDITION PROBLEM -----
      IANS=0
      DO 100 I=1,NUMS
      IANS=IANS+NUMBER(I)
  100 CONTINUE
      IF(IANS.NE.ANSWER) GOTO 150
C
C     ----- INDICATE ANSWER IS CORRECT -----
  105 ENCODE(24,110,ITXT)
  110 FORMAT('YOUR ANSWER IS CORRECT      ')
      CALL TEXTO(IDEV,ITXT,6,36,10,2,1,IER)
      IF(IER.NE.0) OUTPUT(101) IER, 'CHECK ADD 1'
      KNTT=KNTT+1
      RETURN
C
C     ----- INDICATE ANSWER IS INCORRECT -----
  150 ENCODE(28,170,ITXT)
```

```
170 FØRMAT('YØUR ANSWER IS NØT CØRRECT  ')
    CALL TEXT8(IDEV,ITXT,7,35,10,2,1,IER)
    IF(IER.NE.0) ØUTPUT(101) IER, 'CHECK ADD 2'
    KNTT=KNTT+1
    ENCØDE(28,180,ITXT)
C
C   ----- DISPLAY THE CØRRECT ANSWER -----
180 FØRMAT('THE CØRRECT ANSWER IS          ')
    CALL TEXT8(IDEV,ITXT,7,39,10,2,1,IER)
    IF(IER.NE.0) ØUTPUT(101) IER, 'CHECK ADD 3'
    KNTT=KNTT+1
    DØ 185 I=1,4
    NIL(I)=77777777B
185 CØNTINUE
    CALL TEXT8(IDEV,NIL,4,39,58,2,1,IER)
    IF(IER.NE.0) ØUTPUT(101) IER, 'NIL'
    KNTT=KNTT+1
    IF(PRØB.NE.4) RETURN
    ICP=65
190 FØRMAT(I4)
    IF(IANS.GE.0.AND.IANS.LT.10) NDX=1
    IF(IANS.GE.10.AND.IANS.LT.100) NDX=2
    IF(IANS.GE.100.AND.IANS.LT.1000) NDX=3
    IF(IANS.GE.1000.AND.IANS.LT.10000) NDX=4
    IF(IANS.GE.10000.AND.IANS.LT.100000) NDX=5
    IF(IANS.GE.100000) NDX=6
192 DØ 194 I=1,NDX
    KANS=IANS-(IANS/10)*10
    IANS=IANS/10
    ENCØDE(4,193,IBUF) KANS
193 FØRMAT(I4)
    CALL TEXT8(IDEV,IBUF,1,39,ICP,2,1,IER)
    IF(IER.NE.0) ØUTPUT(101) IER, 'KANS'
    KNTT=KNTT+1
    ICP=ICP-2
```

```
  194 CONTINUE
      IF(PROB.NE.4) RETURN
C
C     ----- DISPLAY CORRECT REMAINDER -----
      ENCODE(4,195,NTXT)
  195 FORMAT('R = ')
      CALL TEXTO(IDEV,NTXT,1,40,80,2,1,IER)
      IF(IER.NE.0) OUTPUT(101) IER, 'CHECK ADD 5'
      KNTT=KNTT+1
      ENCODE(4,190,NTXT) MAIN
      CALL TEXTO(IDEV,NTXT,1,40,88,2,1,IER)
      IF(IER.NE.0) OUTPUT(101) IER, 'CHECK ADD 6'
      KNTT=KNTT+1
      RETURN
C
C     ----- SUBTRACTION PROBLEM -----
  200 IANS=NUMBER(1)-NUMBER(2)
      IF(IANS.EQ.ANSWER) GOTO 105
      GOTO 150
C
C     ----- MULTIPLICATION PROBLEM -----
  300 IANS=NUMBER(1)*NUMBER(2)
      IF(IANS.EQ.ANSWER) GOTO 105
      GOTO 150
C
C     ----- DIVISION PROBLEM -----
  400 IANS=NUMBER(1)/NUMBER(2)
      MAIN=NUMBER(1)-NUMBER(2)*IANS
      IF(IANS.EQ.ANSWER.AND.MAIN.EQ.REMAIN) GOTO 105
      GOTO 150
      END
```

```
C
C          SUBRØUTINE TØ DEMØNSTRATE TØ THE USER THE PRØPER
C          SØLUTIØN TØ ADDITIØN, SUBTRACTIØN, NULTIPLICATIØN, AND
C          DIVISIØN PRØBLEMS IN A STEP BY STEP MANNER.
C
C
      SUBRØUTINE DEMØ
      INTEGER PRØB,NUMS,RANGE SØRL,REMAIN,CB,ANSWER
      GLØBAL PRØB,NUMS,RANGE,CB,KIGITS(10),SØRL,REMAIN,NUMBER(10),INTERM
     C(10),IDEV,ANSWER,ITDIR(50),IGDIR(50),KNTT,KNTG
      DIMENSIØN ITXT13(13),LINE(5),ITXT8(8),ITXT18(18),ITXT3(3)
C
C     ----- INITIALIZE LØCAL VARIABLES -----
      NULL=77777777B
      KARY=0
      DØ 100 I=1,50
      ITDIR(I)=0
      IGDIR(I)=0
  100 CØNTINUE
      CALL DTINIT(IDEV,ITDIR,50,IER)
      IF(IER.NE.0) ØUTPUT(101) IER, 'DEMØ 1'
      CALL DGINIT(IDEV,IGDIR,50,IER)
      IF(IER.NE.0) ØUTPUT(101) IER, 'DEMØ 2'
      KNTT=0
      KNTG=0
C
C Ø   ----- DISPLAY EXPLANATIØN AND INSTRUCTIØNS -----
      ENCØDE(52,200,ITXT13)
  200 FØRMAT('YØUR LAST PRØBLEM WAS SØLVED INCØRRECTLY              ')
      CALL TEXTØ(IDEV,ITXT13,13,2,1,1,1,IER)
      IF(IER.NE.0) ØUTPUT(101) IER, 'DEMØ 3'
      KNTT=KNTT+1
      ENCØDE(52,300,ITXT13)
  300 FØRMAT('THE SAME PRØBLEM WILL NØW BE SØLVED FØR YØU CØRRECTLY')
      CALL WAIT(2)
```

```
      CALL TEXTØ(IDEV,ITXT13,13,4,1,1,1,IER)
      IF(IER.NE.0) ØUTPUT(101) IER, 'DEMØ 4'
      KNTT=KNTT+1
      ENCØDE(52,400,ITXT13)
  400 FØRMAT('WATCH EACH STEP CLØSELY TØ SEE WHAT YØU DID WRØNG    ')
      CALL WAIT(2)
      CALL TEXTØ(IDEV,ITXT13,13,6,1,1,1,IER)
      IF(IER.NE.0) ØUTPUT(101) IER, 'DEMØ 5'
      KNTT=KNTT+1
      IF(PRØB.GT.3) GØTØ 4000
C
C     ----- DISPLAY THE PRØBLEM -----
      ENCØDE(4,500,ITXT) NUMBER(1)
  500 FØRMAT(I4)
      CALL TEXTØ(IDEV,ITXT,1,20,50,2,1,IER)
      IF(IER.NE.0) ØUTPUT(101) IER, 'DEMØ 6'
      KNTT=KNTT+1
      ENCØDE(4,500,ITXT) NUMBER(2)
      CALL TEXTØ(IDEV,ITXT,1,21,50,2,1,IER)
      IF(IER.NE.0) ØUTPUT(101) IER, 'DEMØ 7'
      KNTT=KNTT+1
      IF(PRØB.EQ.1) ENCØDE(4,600,ITXT)
      IF(PRØB.EQ.2) ENCØDE(4,700,ITXT)
      IF(PRØB.EQ.3) ENCØDE(4,800,ITXT)
  600 FØRMAT('   +')
  700 FØRMAT('   -')
  800 FØRMAT('   X')
      CALL TEXTØ(IDEV,ITXT,1,21,43,2,1,IER)
      IF(IER.NE.0) ØUTPUT(101) IER, 'DEMØ 8'
      KNTT=KNTT+1
      LINE(1)=IHEAD(0,4)
      LINE(2)=IPACK(-0.05,-0.1,0)
      LINE(3)=IPACK(0.3,-0.1,1)
      LINE(4)=0
      LINE(5)=0
```

```
      KNTG=KNTG+1
      CALL GRAPHO(IDEV,LINE,5,KNTG,IER)
      IF(IER.NE.0) OUTPUT(101) IER, 'DEMO 9'
      CALL WAIT(10)
C
C     ----- SOLVE THE PROBLEM -----
      ICP=52
      NBR1=NUMBER(1)
      NBR2=NUMBER(2)
      MAXDIG=KIGITS(1)
      IF(KIGITS(2).GT.MAXDIG) MAXDIG=KIGITS(2)
      IF(PROB.GT.1) GOTO 2000
C
C     ----- HANDLE ADDITION PROBLEMS -----
      DO 1000 I=1,MAXDIG
      N1=NBR1-(NBR1/10)*10
      NBR1=NBR1/10
      N2=NBR2-(NBR2/10)*10
      NBR2=NBR2/10
      NANS=N1+N2+KARY
      IF(NANS.GE.10) GOTO 900
C
C     ----- NO CARRY REQUIRED -----
      ENCODE(4,500,ITXT) NANS
      CALL TEXTO(IDEV,ITXT,1,23,(ICP-2*I),2,1,IER)
      IF(IER.NE.0) OUTPUT(101) IER, 'DEMO 10'
      KNTT=KNTT+1
      KARY=0
      CALL WAIT(5)
      GOTO 1000
C
C     ----- CARRY REQUIRED -----
  900 NX=NANS-(NANS/10)*10
      ENCODE(4,500,ITXT) NX
      CALL TEXTO(IDEV,ITXT,1,23,(ICP-2*I),2,1,IER)
```

```
      IF(IER.NE.0) OUTPUT(101) IER, 'DEMO 11'
      KNTT=KNTT+1
      CALL WAIT(5)
      KARY=(NANS-NX)/10
      ENCODE(4,500,ITXT) KARY
      CALL TEXTO(IDEV,ITXT,1,19,(ICP-2*I+1),1,1,IER)
      IF(IER.NE.0) OUTPUT(101) IER, 'DEMO 12'
      KNTT=KNTT+1
      CALL WAIT(5)
 1000 CONTINUE
      GOTO 5000
 2000 IF(PROB.GT.2) GOTO 3000
C
C     ----- HANDLE SUBTRACTION PROBLEMS -----
      DO 2500 I=1,MAXDIG
      N1=NBR1-(NBR1/10)*10
      NBR1=NBR1/10
      N2=NBR2-(NBR2/10)*10
      NBR2=NBR2/10
      NANS=N1-N2
      IF(NANS.LT.0) GOTO 2200
C
C     ----- NO BORROW REQUIRED -----
      ENCODE(4,500,ITXT) NANS
      CALL WAIT(5)
      CALL TEXTO(IDEV,ITXT,1,23,(ICP-2*I),2,1,IER)
      IF(IER.NE.0) OUTPUT(101) IER, 'DEMO 17'
      KNTT=KNTT+1
      GOTO 2500
C
C     ----- BORROW REQUIRED -----
 2200 ENCODE(12,2300,ITXT3)
 2300 FORMAT('           /')
      CALL WAIT(5)
      CALL TEXTO(IDEV,ITXT3,3,20,(34-2*I),2,1,IER)
```

```
      IF(IER.NE.0) ØUTPUT(101) IER, 'DEMØ 18'
      KNTT=KNTT+1
      NX=NBR1-(NBR1/10)*10-1
      ENCØDE(4,500,ITXT) NX
      CALL WAIT(5)
      CALL TEXT8(IDEV,ITXT,1,19,(ICP-2*I+1),1,1,IER)
      IF(IER.NE.0) ØUTPUT(101) IER, 'DEMØ 19'
      KNTT=KNTT+1
      N1=N1+10
      NBR1=NBR1-1
      CALL WAIT(5)
      ENCØDE(4,500,ITXT) N1
      CALL TEXT8(IDEV,ITXT,1,19,(ICP-2*I+3),1,1,IER)
      IF(IER.NE.0) ØUTPUT(101) IER, 'DEMØ 21'
      KNTT=KNTT+1
      NANS=N1-N2
      ENCØDE(4,500,ITXT) NANS
      CALL WAIT(5)
      CALL TEXT8(IDEV,ITXT,1,23,(ICP-2*I),2,1,IER)
      IF(IER.NE.0) ØUTPUT(101) IER, 'DEMØ 20'
      KNTT=KNTT+1
 2500 CØNTINUE
      GØTØ 5000
C
C     ----- HANDLE MULTIPLICATIØN PRØBLEMS -----
 3000 DØ 3050 I=1,10
      INTERM(I)=0
 3050 CØNTINUE
      LN=23
      KARY=0
C
C     ----- CALCULATE AND DISPLAY INTERMEDIATE RESULTS -----
      DØ 3500 I=1,KIGITS(2)
      N2=NBR2-(NBR2/10)*10
      INT=NBR2 / 10
```

```
      NBR2=INT
C
C     ----- DISPLAY INDIVIDUAL DIGITS ØF INTERMEDIATE RESULTS -----
      DØ 3300 J=1,KIGITS(1)
      N1=NBR1-(NBR1/10)*10
      NBR1=NBR1/10
      NX=N1*N2+KARY
      IF(NX.GE.10) GØTØ 3100
C
C     ----- NØ CARRY REQUIRED -----
      INTERM(I)=INTERM(I)+NX*10**(J-1)
      ENCØDE(4,500,ITXT) NX
      CALL WAIT(5)
      CALL TEXTØ(IDEV,ITXT,1,LN,(ICP-2*J),2,1,IER)
      IF(IER.NE.0) ØUTPUT(101) IER, I, J, 'DEMØ 22'
      KNTT=KNTT+1
      KARY=0
      GØTØ 3300
C
C     ----- CARRY REQUIRED -----
 3100 NANS=NX-(NX/10)*10
      INTERM(I)=INTERM(I)+NANS*10**(J-1)
      NX=NX/10
      ENCØDE(4,500,ITXT) NANS
      CALL WAIT(5)
      CALL TEXTØ(IDEV,ITXT,1,LN,(ICP-2*J),2,1,IER)
      IF(IER.NE.0) ØUTPUT(101) IER,I,J, 'DEMØ 23'
      KNTT=KNTT+1
      KARY=NX
 3300 CØNTINUE
      IF(KARY.EQ.0) GØTØ 3400
C
C     ----- DISPLAY LAST CARRY -----
      ENCØDE(4,500,ITXT) KARY
      CALL WAIT(5)
```

```
      CALL TEXTO(IDEV,ITXT,1,LN,(ICP-2*J),2,1,IER)
      IF(IER.NE.0) OUTPUT(101) IER, 'DEMO 26'
      KNTT=KNTT+1
      INTERM(I)=INTERM(I)+KARY*10**(J-1)
 3400 LN=LN+1
      ICP=ICP-2
      KARY=0
      NOR1=NUMBER(1)
 3500 CONTINUE
C
C     ----- DISPLAY ADDITION LINE -----
      IF(KIGITS(2).EQ.1) GOTO 5000
      LINE(2)=IPACK(-0.1,-0.25,0)
      LINE(3)=IPACK(0.25,-0.25,1)
      IF(KIGITS(2).EQ.2) GOTO 3550
      LINE(2)=IPACK(-0.1,-0.3,0)
      LINE(3)=IPACK(0.25,-0.3,1)
 3550 LINE(4)=0
      LINE(5)=0
      KNTG=KNTG+1
      CALL WAIT(5)
      CALL GRAPHO(IDEV,LINE,5,KNTG,IER)
      IF(IER.NE.0) OUTPUT(101) IER, 'DEMO 24'
      DO 3600 I=1,KIGITS(2)
      INTERM(I)=INTERM(I)*10**(I-1)
 3600 CONTINUE
      NANS=0
C
C     ----- ADD TO OBTAIN FINAL ANSWER -----
      DO 3700 I=1,10
      NANS=NANS+INTERM(I)
 3700 CONTINUE
C
C     ----- DISPLAY FINAL ANSWER -----
      LN=LN+1
```

```
      ICP=52
      DO 3800 I=1,10
      NX=NANS-(NANS/10)*10
      NANS=NANS/10
      IF(NX.EQ.0.AND.NANS.EQ.0) GOTO 5000
      ENCODE(4,500,ITXT) NX
      CALL WAIT(5)
      CALL TEXTO(IDEV,ITXT,1,LN,(ICP-2*I),2,1,IER)
      IF(IER.NE.0) OUTPUT(101) IER, 'DEMO 25'
      KNTT=KNTT+1
 3800 CONTINUE
      GOTO 5000
C
C     ----- HANDLE DIVISION PROBLEMS -----
 4000 ENCODE(4,500,ITXT) NUMBER(2)
C
C     ----- DISPLAY THE DIVISOR -----
      CALL TEXTO(IDEV,ITXT,1,16,43,2,1,IER)
      IF(IER.NE.0) OUTPUT(101) IER, 'DEMO 26'
      KNTT=KNTT+1
      ICP=45+2*KIGITS(1)
      ENCODE(4,500,ITXT) NUMBER(1)
C
C     ----- DISPLAY THE DIVIDEND -----
      CALL TEXTO(IDEV,ITXT,1,16,ICP,2,1,IER)
      IF(IER.NE.0) OUTPUT(101) IER, 'DEMO 27'
      KNTT=KNTT+1
C
C     ----- DISPLAY THE DIVISION LINES -----
      LINE(1)=IHEAD(0,4)
      LINE(2)=IPACK(.05,.22,0)
      LINE(3)=IPACK(.05,.30,1)
      X=0
      DO 4100 I=1,KIGITS(1)
      X=X+0.10
```

```
4100 C0NTINUE
     X=X+0.05
     LINE(4)=IPACK(X,0.30,1)
     LINE(5)=0
     KNTG=KNTG+1
     CALL GRAPH0(IDEV,LINE,5,KNTG,IER)
     IF(IER.NE.0) 0UTPUT(101) IER, 'DEM0 28'
C
C    ----- FIND SEPARATE DIGITS 0F DIVIDEND -----
     D0 4200 I=1,5
     LINE(I)=0
4200 C0NTINUE
     NBR1=NUMBER(1)
     D0 4300 I=1,KIGITS(1)
     LINE(KIGITS(1)-I+1)=NBR1-(NBR1/10)*10
     N=NBR1/10
     NBR1=N
4300 C0NTINUE
C
C    ----- S0LVE AND DISPLAY THE S0LUTI0N -----
     ICP=47
     N=0
     D0 4400 I=1,KIGITS(1)
     NDX=I
     N=N*10+LINE(I)
     NX=N / NUMBER(2)
     IF(NX.GT.0) G0T0 4500
     ICP=ICP+2
4400 C0NTINUE
4500 ENC0DE(4,500,ITXT) NX
     CALL WAIT(5)
     CALL TEXT0(IDEV,ITXT,1,15,ICP,2,1,IER)
     IF(IER.NE.0) 0UTPUT(101) IER, 'DEM0 29'
     KNTT=KNTT+1
     KMUL=NX*NUMBER(2)
```

```
      KDIF=N-KMUL
      IF(NDX.LT.KIGITS(1)) GØTØ 4700
C
C     ----- DISPLAY THE REMAINDER -----
 4550 ENCØDE(4,4600,ITXT)
 4600 FØRMAT(' R =')
      CALL WAIT(5)
      CALL TEXTØ(IDEV,ITXT,1,20,50,2,1,IER)
      IF(IER.NE.0) ØUTPUT(101) IER, 'DEMØ 30'
      KNTT=KNTT+1
      ENCØDE(4,500,ITXT) KDIF
      CALL TEXTØ(IDEV,ITXT,1,20,55,2,1,IER)
      IF(IER.NE.0) ØUTPUT(101) IER, 'DEMØ 31'
      KNTT=KNTT+1
      GØTØ 5000
 4700 NDX=NDY+1
      N=KDIF*10+LINE(NDX)
      NX=N / NUMBER(2)
      IF(NX.EQ.0.AND.NDX.GE.KIGITS(1)) GØTØ 4800
      IF(NX.EQ.0) GØTØ 4700
      ICP=ICP+2
      CALL WAIT(5)
      ENCØDE(4,500,ITXT) NX
      CALL TEXTØ(IDEV,ITXT,1,15,ICP,2,1,IER)
      IF(IER.NE.0) ØUTPUT(101) IER, 'DEMØ 32'
      KNTT=KNTT+1
      KMUL=NX*NUMBER(2)
      KDIF=N-KMUL
      NDX=NDX+1
      IF(NDX.GT.KIGITS(1)) GØTØ 4550
      NDX=NDX-1
      GØTØ 4700
 4800 K=KDIF*10+LINE(NDX)
      KDIF=K
      GØTØ 4550
```

```
C
C     ----- DISPLAY FURTHER INSTRUCTIØNS -----
 5000 ENCØDE(32,6000,ITXT8)
 6000 FØRMAT('NØW LØØK AT THE PRØBLEM AGAIN    ')
      CALL WAIT(2)
      CALL TEXTØ(IDEV,ITXT8,8,38,1,1,1,IER)
      IF(IER.NE.0) ØUTPUT(101) IER, 'DEMØ 13'
      KNTT=KNTT+1
      ENCØDE(72,7000,ITXT18)
 7000 FØRMAT('IF YØU UNDERSTAND THE MISTAKE YØU MADE TYPE YES AND HIT TH
     CE RETURN KEY. ')
      CALL WAIT(2)
      CALL TEXTØ(IDEV,ITXT18,18,39,1,1,1,IER)
      IF(IER.NE.0) ØUTPUT(101) IER, 'DEMØ 14'
      KNTT=KNTT+1
      ENCØDE(72,8000,ITXT18)
 8000 FØRMAT('IF YØU DØNT UNDERSTAND YØUR MISTAKE TYPE NØ AND HIT THE RE
     CTURN KEY        ')
      CALL TEXTØ(IDEV,ITXT18,18,40,1,1,1,IER)
      IF(IER.NE.0) ØUTPUT(101) IER, 'DEMØ 15'
      KNTT=KNTT+1
C
C     ----- WAIT FØR USER RESPØNSE -----
      CALL TEXTR(IDEV,NULL,1,40,82,2,1,IER)
      IF(IER.NE.0) ØUTPUT(101) IER, 'DEMØ 16'
      KNTT=KNTT+1
 9000 IF(MØD(ITDIR(KNTT),8).EQ.0) GØTØ 5000
      RETURN
      END
```

BIBLIOGRAPHY

1. Bryan, G. L., "Computers and Education," Computers and Automation, v. 18, p. 16-19, March 1969.

2. Bushnell, D. D. and Allen, D. W., The Computer in American Education, John Wiley and Sons, Inc., 1967.

3. Coulson, J. E., Programmed Learning and Computer-Based Instruction, John Wiley and Sons, Inc., 1962.

4. Dunca, E. R., and others, Modern School Mathematics Structure and Use, California State Department of Education, 1970.

5. Entelek, Incorporated, Project Number 154-254, Computer-Assisted Instruction, A Survey of the Literature, by A. E. Hickey and J. M. Newton, January 1967.

6. Ferguson, R. L., "Computer Assistance for Individualizing Instruction," Computers and Automation, v. 19, p. 27-29, March 1970.

7. Gerald, R. W., Computers and Education, McGraw-Hill, 1967.

8. Long, H. S. and Schwartz, H. A., "Instruction by Computer," Datamation, v. 12, p. 73-87, September 1966.

9. Oettinger. A. C., Run, Computer, Run, Harvard Press, 1969.

10. Skinner, B. F., The Technology of Teaching, Appleton-Century-Crofts, 1968.

11. Stolurow, L. M., The Schools and the Challenge of Innovation, McGraw-Hill, 1969.

12. Suppes, P., "The Uses of Computers in Education," Scientific American, v. 215, p. 207-220, September 1966.

13. Unwin, D. and Leedham, J., Aspects of Educational Technology, Methuen, 1967.

14. Zinn, K. L., "Instructional Uses of Interactive Computer Systems," Datamation, v. 14, p. 22-27, September 1968.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	LTJG R. C. Bolles, Code 53Bq Department of Mathematics Naval Postgraduate School Monterey, California 93940	1
4.	LCDR Steven Selby Abrams 917 White Plains Road Charlotte, North Carolina 28213	1
5.	LT Ronald D. DeLaura, Code 52Dl Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)

Naval Postgraduate School
Monterey, California 93940

2a. REPORT SECURITY CLASSIFICATION

Unclassified

2b. GROUP

3. REPORT TITLE

A Complete Interactive Graphical Computer-Aided Instruction System

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Master's Thesis; June 1971

5. AUTHOR(S) (First name, middle initial, last name)

Steven Selby Abrams

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
June 1971	72	14

8a. CONTRACT OR GRANT NO.

b. PROJECT NO.

c.

d.

9a. ORIGINATOR'S REPORT NUMBER(S)

9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES

12. SPONSORING MILITARY ACTIVITY

Naval Postgraduate School
Monterey, California 93940

13. ABSTRACT

The use of interactive graphics in computer-aided instruction systems is discussed with emphasis placed on two requirements of such a system. The first is the need to provide the teacher with a useful tool with which to design and modify teaching sessions tailored to the individual needs and capabilities of the students. The second is the requirement to provide for sufficiently flexible interaction with the student during the actual teaching session. These concepts are implemented in a system which maximizes the use of interactive graphics. Facilities available in the system present material to the student in a dynamic, graphical fashion which leads to better understanding and a more enjoyable learning experience. The implemented system, while not yet complete, indicates that a complete working system of this type is feasible and would prove itself to be a valuable educational tool.

KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
computer-aided instruction computer graphics computer-assisted learning						

DD FORM 1 NOV 65 1473 (BACK)

N 0101-807-6821

Thesis 128204
A17 Abrams
c.1 A complete interactive graphical computer-aided instruction system.